Graph algorithm for deterministic routing to ensure latence in cloud RAN networks

MG, DB, CC, OM, YS

January 16, 2017

1 Introduction

TODO: A REPRENDRE POUR LA JUSTIFICTION RESEAU AVEC LE DETERMINISTIC ROUTING (OM?) ET INSERER LA PROBLMATIQUE GRAPHE (DB) In this article, we will work on latency constraints in fronthaul. Thus, this type of architecture confronts the problem of controlling the latency in the transfer process. Low latency is considered critical for the 5G, in particular for the deployment of the C-RAN approach (allowing time constraints like HARQ to be fulfilled over non dedicated networks), or to reach E2E expected latency from 1 to 10ms (depending on targeted services). One specificity in the C-RAN context is not only the latency constraint, but also the periodicity of the data transfer between RRH and BBU. New scheduling and routing paradigms and new technologies have to be considered to guarantee delay constrained periodic data transfers. Dynamical optical bypass and dynamical management of the emission should be considered to guarantee latency constraints. This is why this study is a contribution to the ANR project N-Green.

N-GREEN proposes a new type of switching/routing node and a specific network architecture exploiting WDM packets thanks to a new generation of optical add/drop multiplexers (WSADM: WDM slotted add/drop multiplexer). These packets having a fixed duration close to 1μs are transported in a transparent way, to better exploit the switching matrix of the node; their headers will be transported over one dedicated wavelength at a lower bit rate, to reduce the physical constraints of the electronic processing and scheduler.

Thus, this subject targets new scheduling and routing paradigms to solve this periodic and delay constrained data transfer. Indeed, one of the most promising approaches relies on the concept of Deterministic Networking (DN) such that one get rid of statistical multiplexing. The traditional queue managements are replaced by time based forwarding. Solutions for Deterministic Networking are under standardization in IEEE 802.1 TSN group, as well at IETF DetNet working group. To make DN working over a network composed of several nodes, it is required to manage the time at which the packets of deterministic paths are crossing each nodes.

Considering a graph, modeling the network topology, and a set of routes from source nodes (modeling connections to the BBU) and destination nodes (modeling the RRH) in this graph, the purpose is to select, for each destination node a route from one source node to it and a periodic routing scheme allowing to periodically sent a packet to each base station without congestion conflicts between all such packets and to insure a minimum latency. In a slotted time model, the aim is here to minimize the duration of the period, with a constraint of the maximum length of routes to be selected. Even if the selected set of routes is given this optimization problem has been shown to be NP-hard. From an algorithmic point of view, the purpose of this project is first, to study the complexity and the approximability of this problem when the length of the routes is small (which corresponds to realistic cases), and secondly, to propose and implement some heuristics to solve this problem on realistic topologies.

The major difficulty of this problem is the periodicity of the process. Indeed, a deterministic sending for the messages between each pair BBU/RRH must not collide with the other messages sent by the others BBU/RRH in the same period, but also in the previous and following periods. This problem may look like wormhole problem [?], very popular few years ago, but here, we want to minimize the time lost in buffers and not just avoid the deadlock, and the wormhole does not treat about the periodicity.

Related works

TODO: IL FAUT REPARTIR DE CE QUI EST DANS LE MEMOIRE DE MAEL, PLUS LES ARTICLES D'ORDONNANCMENT DE TRAINS. JE (DB) M'EN CHARGE.

2 Model and problem description

2.1 Definitions

We consider a symmetric directed graph $G = (V, A)$ modeling a slotted network. Each arc (u, v) in A is characterized by an integer delay $Dl(u, v) \geq 1$ representing the number of slots from u to v on this arc. Note that for any arc (u, v), $DL(u, v) = DL(v, u)$.

A route r in G is a sequence of consecutive arcs $a_0, \ldots, a_{k-1}$, with $a_i = (u_i, u_{i+1}) \in A$. The *latency* of a vertex u_i in r, with $i \geq 1$, is defined by

$$\lambda(u_i, r) = \sum_{0 \leq j < i} Dl(a_j)$$

We also define $\lambda(u_0, r) = 0$. The latency of the route r is defined by $\lambda(r) = \lambda(u_k, r)$. A routing function $\mathcal{R}$ in G associates a route from u to v for any couple of different vertices $< u, v >$ in G.

Let $\mathcal{C}$ be an assignment in G, i.e., set of couples of different vertices of G. We denote by $\mathcal{R}_{\mathcal{C}}$ the set of routes $\mathcal{R}(u,v)$ for any $< u, v >$ in $\mathcal{C}$.

Consider now a positive integer P called **period**. A **P-periodic affectation** of $\mathcal{C}$ in $(G, \mathcal{R})$ consists in a set $\mathcal{M} = (m_0, \ldots, m_{c-1})$ of c integers that we call **offset**, with c the cardinal of $\mathcal{C}$. Indeed, time is consider as consecutive periods of P slots each and the number m_i represents the first slot number used by the route $r_i \in \mathcal{R}_{\mathcal{C}}$ at its source. We define the first time slot at which a message reaches any vertex v in this route by

$$t(v, r_i) = m_i + \lambda(v, r_i) \mod P.$$

Let us call $[t(v, r_i)]$ the values of the time slots used by a route r_i in such a vertex v. Those values are forming a continuous set of values starting at $t(v, r_i)$ mod P and ending at $t(v, r_i) + \tau \mod P$. A P-periodic affectation must have no **collision** between two routes in $\mathcal{R}_{\mathcal{C}}$, that is $\forall r_i, r_j \in \mathcal{R}_{\mathcal{C}}, i \neq j$, with τ the size (in number of consecutive slots) of each message that must be periodically sent on each route of $\mathcal{R}_{\mathcal{C}}$, we have

$$[t(u, r_i)] \cap [t(u, r_j)] = \emptyset.$$

The main theoretical problem we have to deal with in this context is the following.

Problem Periodic Routes Assignment (PRA)
Input: graph $G = (V, A)$, set $\mathcal{C}$ of couples of vertices, routing function $\mathcal{R}$, integer P.
Question: does there exist a P-periodic affectation of $\mathcal{C}$ in $(G, \mathcal{R})$?

We deal in next section with complexity of Problem PRA.

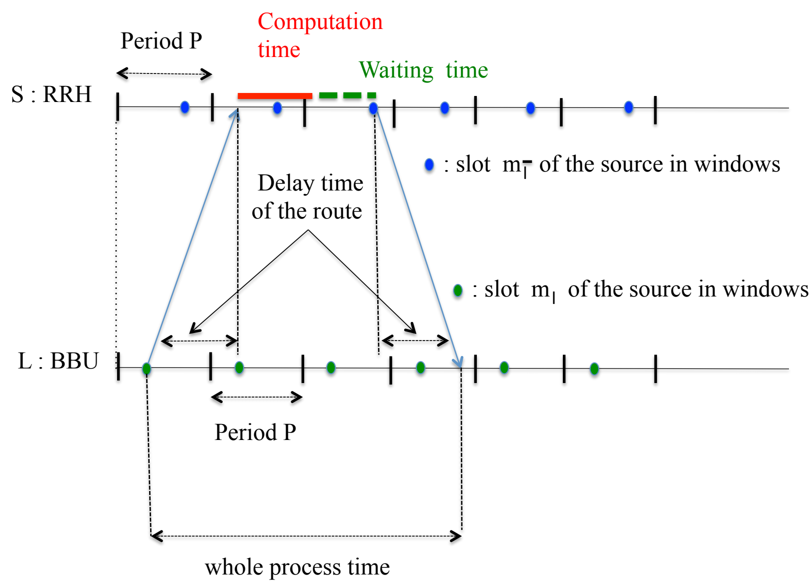

Figure 1: Complete process for a leaf in L.

In the context of cloud-RAN applications, we consider here in digraph $G = (V, A)$ modeling the target network two disjoint subsets of vertices S and L, where S is the set of BBU and L is the set of RRH. We consider a **matching** defined as an application $\rho : S \to L$. We consider a P-periodic affectation $\mathcal{M}$, associating respectively integers m_l and $m_{\bar{l}}$ to couples $(l, \rho(l))$ and $(\rho(l), l)$ for all $l \in L$. The process realized periodically (i.e. initiated in each window of size P) for each leaf $l \in L$ is the following one (see Figure **??**). First, a message of τ slots is sent from l to $\rho(l)$ on $\mathcal{R}(l, \rho(l))$ at slot m_l in the current window. After receiving this message, $\rho(l)$ computes it during a time equal to θ slots. Then, a message of τ slots containing the computed result is sent back from $\rho(l)$ to l on $\mathcal{R}(\rho(l), l)$, at the first occurrence of step $m_{\bar{l}}$ in a window (i.e., in the current window at the end of computation or the next one). We denote by $\omega(l)$ the **waiting time**of l, i.e., the number of slots between the end of the computation time in $\rho(l)$ and the first occurrence of step $m_{\bar{l}}$ in a window. Thus, the whole proccess time for l is equal to

$$PT(l) = \lambda(\mathcal{R}(l, \rho(l))) + \theta + \omega(l) + \lambda(\mathcal{R}(\rho(l), l)).$$

Let us denote by $\mathcal{C}_\rho$ the set of couples $< l, \rho(l) >$ and $< \rho(l), l >$, for any $l \in L$. Consider a P-periodic affectation $\mathcal{M}$ of $\mathcal{C}_\rho$ in $(G, \mathcal{R})$. The maximum process time of $\mathcal{M}$ is equal to equal to $MT(\mathcal{M}) = \max_{l \in L} PT(l)$. Thus, we have to deal with the following problem.

Problem Periodic Assignment for Low Latency(PALL)
Input: a digraph G, a matching ρ of a set S into a set L in G, a routing function $\mathcal{R}$, a period P, an integer T_{max}.
Question: does there exist a P-periodic affectation $\mathcal{M}$ of $\mathcal{C}_\rho$ in $(G, \mathcal{R})$ such that $MT(\mathcal{M}) \leq T_{max}$?

The related optimisation problem we will focus on consists in minimizing $MT(\mathcal{M})$. Note that in the context of cloud-RAN networks, we consider $P = 1ms$, $\theta = 2.6ms$ and T_{max} must be less or equal to $3ms$.

3 Complexity results

3.1 Complexity of PRA and PALL

Consider an instance of Problem PRA, i.e., a digraph $G = (V, A)$, an assignment $\mathcal{C}$, a routing function $\mathcal{R}$ and a period P. The *conflict depth* of an assignment $\mathcal{C}$ is the number of arcs crossed by at least two routes in $\mathcal{R}_\mathcal{C}$; in other words it is the number of potential conflicts between these routes. The *load* of an assignment is the maximal number of routes sharing a same arc (i.e., the congestion of this arc). It is clear that a P-periodic affectation must satisfy that P is larger or equal to the load.

We give two alternate proofs that PRA is NP-complete. The first one works for conflict depth 2 and is minimal in this regards since we later prove that for conflict depth one, it is easy to solve PRA. The second one reduces the problem to graph coloring and implies inapproximability.

Proposition 1. *Problem PRA is* NP*-complete, for a routing with conflict depth two.*

Proof. Let $H = (V, E)$ be a graph and d its maximal degree. We want to determine whether H is edge-colorable with d or $d+1$ colors. We reduce this edge coloring problem to PRA. We define from H an instance $G, \mathcal{C}, \mathcal{R}, P$ as follows. For each $v \in V$ there are two vertices in G connected by an arc (v_1, v_2) and none of these arcs are incident. The delay of these arcs is equal to 1. The set $\mathcal{C}$ is made of couples of vertices $< s_{u,v}, l_{u,v} >$ for each edge $[u, v]$ in H and the routes associated is $\mathcal{R} = s_{u,v}, u_1, u_2, v_1, v_2, l_{u,v}$. The delays of arcs $(s_{u,v}, u_1)$ and $(v_2, l_{u,v})$ are equal to $d(d+1) - 1$.

Let ϕ be an edge coloring with k colors of H. We can build a k-periodic affectation of $\mathcal{C}$ in $(G, \mathcal{R})$ by assigning to the source $s_{u,v}$ of each route an offset of $\phi(s_{u,v})$. Indeed, if two routes r_1 and r_2 in $\mathcal{R}_\mathcal{C}$ share the same edge, say (v_1, v_2) then they represent two edges e_1 and e_2 of H incident to the vertex v. Therefore $\lambda(v_1, r_1) = \phi(e_1) \mod k$ because the delays of the edges before v_1 sum to $d(d+1)$ or $2(d(d+1))$ which are equal to 0 modulo k since $k = d$

or $k = d + 1$. Thus $\lambda(v_1, r_1) - \lambda(v_1, r_2) \mod k = \phi(e_1) - \phi(e_2) \mod k$ and $\lambda(v_1, r_1) - \lambda(v_1, r_2) \mod k > 0$ since $\phi(e_1) \neq \phi(e_2)$.

Now consider a k-periodic affectation of $\mathcal{C}$ in $(G, \mathcal{R})$. For each $[u, v]$ in H, we define $\phi(u, v)$ to be the offset of the route beginning at $s_{u,v}$ in $\mathcal{R}_{\mathcal{C}}$. For the same reasons as in the last paragraph, ϕ is an edge coloring with k colors. Therefore we have reduced edge coloring which is NP-hard [?] to PRA which concludes the proof. □

Solving Problem PALL implies to solve PRA for an assignment $\mathcal{C}_\rho$ defined on S and L (see Section 2). Thus, we have the following corollary :

Corollary 1. *Problem PALL is NP-Hard.*

As another corollary of the previous proposition, given G, $\mathcal{R}$, P and two disjoint subsets S and L in G, the problem of knowing if there is a matching ρ from L in S such that the answer to problem PRA for instance $G, \mathcal{C}_\rho, \mathcal{R}, P$ is NP-hard since checking a potential feasible solution for this problem implies to solve PRA.

TODO: SOMMES NOUS SURS DE CES COROLLAIRES (YS et CC)????

Consider now MIN-PRA as the minimisation problem related to PRA, consisting in minimizing the peril P.

Theorem 1. *Problem MIN-PRA cannot be approximate within a factor $n^{1-o(1)}$ unless $\mathsf{P} = \mathsf{NP}$ even when the load is two and n is the number of vertices.*

Proof. We reduce PRA to graph coloring. Let H be a graph instance of the k-coloring problem. We define G in the following way: for each vertex v in H, there is a route r_v in G. Two routes r_v and r_u share an arc if and only if $[u, v]$ is an edge in H; this arc is the only one shared by these two routes. We put delays inbetween shared arcs in a route so that there is a delay k between two such arcs.

As in the previous proof, a k-coloring of H gives a k-periodic affectation in G and conversely. Therefore if we can approximate the minimum value of P within a factor f, we could approximate the minimal number of colors needed to color a graph within a fator f, by doing the previous reduction for all possible k. The proof follows from the hardness of approximability of finding a minimal coloring [?]. □

In particular, this reduction shows that even with small maximal load, the minimal period can be large.

Proposition 2. *The solution to MIN-PRA is either the load or the load plus one. Moreover, a solution of load plus one can be built in polynomial time.*

Proof. First we define an alerning path and how it characterizes an optimal solution of PRA. Let v be a vertex of degree d in the congestion graph (to be defined). We assume that the coloring of the edges is optimal (to define in the case of a congestion graph). The color α is not used in it neighborood. For every edge of color β from this vertex we build an $\alpha - \beta$ path, that is a maximal path $u_0, u_1, \dots, u_l$ so that (u_0, u_1) is of color $\beta, \beta + \lambda(u_0, u_1)$, then (u_1, u_2) is of color $\alpha + \lambda(u_0, u_1), \alpha + \lambda(u_0, u_2)$ and so on (changer λ car on ne suit pas des routes).

A maximal $\alpha - \beta$ path must be a cycle or the coloring is not minimal. $\square$

TODO: FINALISER CETTE DERNIERE PROPOSTION S'I Y A LIEU (YS ET CC)

3.2 A polynomial case about PRA

We now study special cases of problem MIN-PRA restricted to a subset. First, we consider in this section instances $(G = (V, A), \mathcal{R}, \mathcal{C}_\rho)$ of MIN-PRA in which the routing function is coherent. A routing function $\mathcal{R}$ is **coherent** if and only if for any couples $< u, v >$ and $< u', v' >$ of nodes in V such that the two routes $\mathcal{R}(u, v)$ and $\mathcal{R}(u', v')$ cross two same vertices x and y in this same order, then the subroutes from x to y in $\mathcal{R}(u, v)$ and $\mathcal{R}(u', v')$ are equal. As a consequence, for each node u in V, the subgraph of G induced by all the routes from u to all the other nodes in G is a tree.

Moreover, we also consider that for any pair of symmetric arcs (u, v) and $v, u)$, routes $\mathcal{R}(u, v)$ and $\mathcal{R}(v, u)$ are also symmetric.

Finally, considering sets S and L, set $\mathcal{C}_\rho$ is obtained from an assignment ρ of S in L such that there are no common arcs for routes originating from different sources (ρ is said **source-free**).

Proposition 3. *Problem MIN-PRA restricted to instances $(G = (V, A), \mathcal{R}, \mathcal{C}_\rho)$ within a coherent routing can be solved in linear time according to the size of A.*

The definition of source-free assignment ensures that an arc cannot belong to two routes in $\mathcal{R}$ originating from different sources in S. Moreover, the definition of a coherent routing ensures that if two routes originate from the same source x, they share the same arcs from x to a given vertex y and cannot share an arc after. This means that $\forall (u, v) \in A$, the arcs (u, v) with the highest load $l_{max} = max(load(u, v))$ are arcs a_0^j sharing a common origin $u_0 \in S$ and a P-periodic affectation for those arcs is a P-periodic affectation for all the arcs in A.

A P-periodic affectation is such that routes on a same arc must be separated by a delay that is strictly superior to an integer $\delta \geq 0$. As a consequence, the minimum size of the period P is equal to $l_{max} \times (\delta + 1)$. This means that that on the arc with the highest load, we can schedule the first route at the moment $m_k = 0$, the second route at a moment $m_{k'} = \delta + 1$ and so on for all l_{max} routes.

Proposition 4. *Giving G and a coherent routing $\mathcal{R}$,finding a source-free assignment ρ such that the period solution to MIN-PRA is minimum over all such assignment can be done in linear time according to the size of A.*

In any source-free assignment, the minimum size of P is obtained by minimizing the highest load of the first arc of any route from a source to a leaf. As all sources in S are connected to all sources vertices in L by a route in $\mathcal{R}$, a simple load balancing allows to obtain a maximum load equal to $max_{load} = \lceil \frac{\mathcal{L}}{|S|} \rceil$. Once the assignment is computed, the minimum value of P is thus equal to $max_{load} \times (\delta + 1)$.

We consider now instances $(G, \mathcal{R}, \mathcal{C}_\rho)$ of MIN-PRA, called **1-instances** in which the routing function is still coherent and ρ is such that there exists one and only on arc crossed by at least 2 routes originated from different sources.

Proposition 5. *Problem MIN-PRA restricted to 1-instances can be solved in linear time according to the size of A.*

Let (a, b)be the arc in G crossed by at least 2 routes originated from different sources, and let $r(a, b)$ be the set of such routes. In order to have a P-periodic affectation, the period P must be large enough for all arcs in $r(a, b)$, thus the minimal size of P is $P = load(a, b) \times (\delta + 1)$. Moreover, this is also the minimum solution for $MIN - PRA$. If we consider that if there exists a $load(a_b)$-periodic affectation $m_i \in \mathcal{M}'$ of each route r_i in $r(a, b)$, then m_j in $\mathcal{M}$ corresponding to r_i is $m_j = m_i - \lambda(u_b) mod P$. As the routing function is coherent, no two routes originating from a same source can use different routes to attain (a, b) thus the P-periodic affectation is valid for any arc $(u, v) \in A$ if it is valid on (a, b).

4 Heuristic approaches to solve PALL

4.1 Network topologies

We will especially focus on three symmetric digraph topologies corresponding to realistic cases of fronthaul network. In those kinds of networks, we can differentiate three kinds of topologies. Each one of them correspond to a real configuration, depending of the distance between the BBU and the RRH.

1. Topology 1: A basic network topology, composed of some base stations, represented by source nodes S, all connected to the same switch, which will be a vertex, connected itself to another vertex, corresponding to a switch, connected to some leave nodes L representing the RRH.

2. Topology 2: A network containing an optical ring, such that some sources nodes be connected to it anywhere, not intersecting themselves before the ring, and some set of leave nodes are also connected at any point of the ring.

3. Topology 3: The general case: A DAG, on which we may restrict parameters as the degree or the number of vertices to represent graphs corresponding to realistic networks.

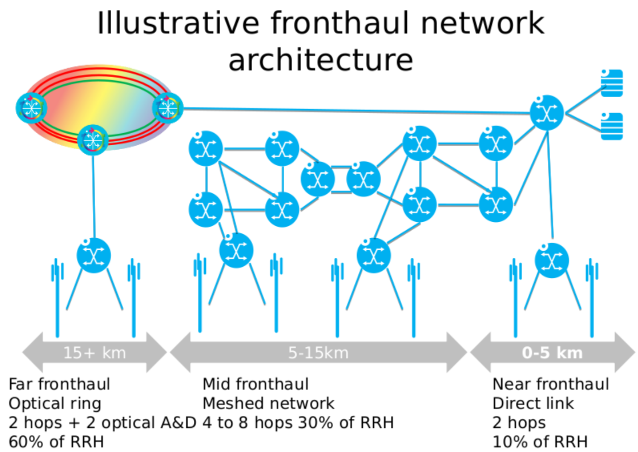

Figure 2: From left to right: Topologies 2, 3 and 1 representing respectively the far, mid and near fronthaul.

4.2 Algorithms

...

4.3 Performance evaluation

...

5 Conclusion

...

6 ANNEXE : MATERIEL SUPPLEMENTAIRE?

6.1 Coloring and P-periodic affectation

We define two characteristic graphs associated to an assignment in G.

Definition 1 (Route Conflict Graph)**.** *The Route Conflict Graph (RCG) of an assignment $\mathcal{C}$ in a digraph $G = (V, A)$ is the graph $G' = (V', E')$ where the vertices of V' are the routes in $\mathcal{R}_{\mathcal{C}}$ and there is an edges between two vertices in V' corresponding to two routes in $\mathcal{R}_{\mathcal{C}}$ if and only if they share a same arc.*

We fix an arbitrary ordering on the elements of $V' = \{v_1, \dots, v_n\}$. We associate a weight to each edge (v_i, v_j). Assume that $i < j$, and that the routes r_i and r_j in $\mathcal{R}_{\mathcal{C}}$ corresponding to v_i and v_j in G' have u as first common vertex. Then the weight of $[v_i, v_j]$ is $\lambda(u, r_i) - \lambda(u, r_j)$.

TODO: FAIRE LE LIEN ENTRE COLLORATION DU RCG ET P-PERIODIC AFFECTATION. NE FAUT-IL PAS INTRODUIRE LA TAILLE DE MESSAGE T DANS LE SYSTEME D'INEQUATIONS? We define $\alpha(u, r_i, r_j) = |\lambda(u, r_i) - \lambda(u, r_j)|$. To the route conflict graph, we can associate a system of inequations representing the constraints that the P-periodic affectation must satisfy.

Definition 2 (Conflict system)**.** *Let G be a weighted RCG, we associate to each vertex v_i of G the variable x_i. The conflict system is the set of inequations $x_i \neq x_j + w(e)$ for $i < j$ and $e = (v_i, v_j)$ edge of G.*

It is simple to see that the conflict system of a assignment has a solution modulo P if and only if the assignment has a P-periodic affectation. This kind of system can be solved by SAT solver or CSP solver and those two methods will be investigated on practical instances. TODO: Donner la formulation pour ces solvers et donner le résultat d'expérience dans une partie indépendante. Les comparer avec un solveur de notre cru avec quelques heuristiques simples.

Definition 3 (Additive coloring)**.** *Le G be a weighted graph, we say that G has an additive coloring with p colors if and only if its associated conflict system has a solution over $\mathbb{Z}/p\mathbb{Z}$. The minimal p for which the system has a solution is the additive chromatic number of G denoted by $\chi_+(G)$.*

Finding an optimal additive coloring is thus the same as solving our problem of finding a P-periodic affectation and is at least as hard as finding an optimal coloring. Because of the constraints on the edges, the additive chromatic number of a graph may be much larger than its chromatic number. When the graph is a clique both additive and regular chromatic number may be the size of the graph. However, we will now prove that for bipartite graphs the chromatic number is two while its additive chromatic number is arbitrary large.

We have the following values obtained by exhaustive search.

TODO: généraliser chi+ aux graphes sans poids en faisant un max ?

Fact 1. *The values we list here are the maximal ones we obtain by weighting bipartite graphs.*

1. $\chi_+(K_{2,2}) = \chi_+(K_{3,3}) = 3$ *with weight* $0, 1$
2. $\chi_+(K_{3,4}) = \chi_+(K_{3,5}) = 3$
3. $\chi_+(K_{3,6}) = 4$
4. $\chi_+(K_{4,4}) = \chi_+(K_{4,5}) = \chi_+(K_{4,6}) = 4$
5. $\chi_+(K_{5,5}) =?$

A nice theoretical question would be to find the way to put weights on a bipartite graph so that its additive chromatic number is maximal. My conjecture is that a $K_{l,l}$ can have an additive chromatic number in $O(l)$. The question is also interesting when we restrict the weights to $0, 1$.

Theorem 2. *There is a weighting of* $K_{l^2,\binom{l}{l^2}}$ *such that* $\chi_+(K_{l^2,\binom{l}{l^2}}) > l$.

Proof. Let V_1, V_2 be the bipartition of the graph we build and let $|V_1| = l^2$. For all $S \subseteq V_1$ with $|S| = l$, we denote by $v_1, \ldots, v_l$ its elements, there is a single element v_S in V_2 which is connected to exactly the elements of S and such that the weight of v_S, v_i is $i - 1$. Because of this construction, V_2 is of size $\binom{l}{l^2}$. Moreover for any additive coloring of the graph we have constructed, a set of l elements in V_1 cannot have all the same color. But by the extended pigeon principle, since there are l^2 elements in V_1 at least l amongst them must have the same color. This prove that the graph we have built cannot have an additive coloring with l colors. □

The theorem can be improved so that the number of colors is logarithmic in the size of the bipartite graph.

Theorem 3. *There is a weighting of* $K_{l^2,\binom{l}{l^2}}$ *such that* $\chi_+(K_{l^2,\binom{l}{l^2}}) > l$.

Proof. Let $\mathcal{F}$ be a family of perfect hash functions from $[l^2]$ to $[l]$. It means that for any subset S of size l of $[l^2]$, there is an hash function $f \in \mathcal{F}$ such that f_S is injective. The construction of the bipartite graph is similar to the previous proof. Let V_1, V_2 be the bipartition of the graph we build and let $V_1 = \{v_1, \ldots, v_{l^2}\}$. For each function $f \in \mathcal{F}$ there is a vertex $v_f \in V_2$ and the weight of (v_i, v_f) is $f(i)$. By [**?**, **?**] there is a family of perfect hash functions of size $2^{O(l)}$ therefore V_2 is of size $2^{O(l)}$. Again by using the pigeon principle and the perfect property of the family of functions, we prove that no additive coloring with l colors is possible. □

TODO: explain the removal of low degree vertices = kernelization + heuristic on the degree Also implement it in the solver

TODO: Comprendre la valeur sur d'autre familles de graphe, notemment celles qu'on peut rencontrer en pratique, par exemple petite tree width includegraphics[scale=0.3]DP-NPA.pdf

Figure 3: Reduction of DPA-NPA into a flow problem

6.2 The disjoint paths NPAC-Problem : DP-NPAC

In this problem, there is a delay constraint that must be satisfied by a assignment. This means that we must remove the routes in $r' \in \mathcal{R}$ where $\lambda(r') > K$.

Proposition 6. *Problem DP-NPAC can be solved in polynomial time according to the size of V.*

In a similar way to problem DP-PRA, the arcs with the highest load can only be the arcs a_0^j sharing a common origin $u_i \in S$. In order to minimize the size of the period P, we have to find a assignment such that the maximum number k_i of routes originating from a same source $u_i \in S$ is minimal. Having found this minimal value k, we face a problem equivalent to the DP-PRA problem.

In order to find a matching of vertices in S with vertices in L such that the maximum number of vertices in L assigned to a vertex $s_i \in S$ is inferior or equal to k, we will transform our problem in a flow problem.

6.2.1 Construction of a flow graph

Let us consider an instance $I = (G = (V, A), S, L, \mathcal{R}, \mathcal{P}, \delta, \mathcal{K}$ and $M)$ of NPA where the routes in $\mathcal{R}$ are only those that respect the delay constraint K. We first construct a complete bipartite graph $G' = (V', A')$ where V' is made of two sets of vertices : vertices V_1' corresponding to S and V_2' corresponding to L. A' is made of all possible arcs from vertices $v_1' \in V_1'$ to vertices $v_2' \in V_2'$, with a capacity 1. We then add to V' a source node S' and a sink node T'. Finally we add to A' all the arcs from S' to each vertex $v_1' \in V_1'$, with a capacity k and all the arcs from each vertex $v_2' \in V_2'$ to T', with a capacity 1, where there is a route $\mathcal{R}(v_1', v_2')$. We have thus obtained a flow graph G' whose size is polynomial in regards to the size of G. We will now compute the maximum flow in G' in order to determine if its size is at least $\mathcal{L}$, in order to be able to connect all the leaves.

We can compute the size of a maximum flow in G' in a polynomial time using a generic flow algorithm as Ford-Fulkerson (it will terminate as arcs capacity are rational numbers). In order to minimize the objective k, we can begin with a value $k = 1$ and use a dichotomic approach to find the minimal value of k for which a maximal flow of size $\mathcal{L}$ exists in G'. The maximum value of k is M. If $k = M$ and the maximum flow value in $G' < \mathcal{L}$, then there is no valid assignment of S in L.

The complexity of minimizing k is thus $0(m^2 n \times log_2 n)$ where $m = |A|$ and $n = |V|$. We obtain in the end a assignment minimizing the maximal number of routes originating from a single source $u_s \in S$ and we are faced with the DP-PRA problem for this instance.