Maximum flow under proportional delay constraint

Pierre Bonami
Aix-Marseille Université, CNRS, LIF UMR 7279, 13000, Marseille, France
IBM ILOG CPLEX, Madrid, Spain
Pierre.Bonami@lif.univ-mrs.fr

Dorian Mazauric
Aix-Marseille Université, CNRS, LIF UMR 7279, 13000, Marseille, France
Dorian.Mazauric@lif.univ-mrs.fr

Yann Vaxès
Aix-Marseille Université, CNRS, LIF UMR 7279, 13000, Marseille, France
Yann.Vaxes@lif.univ-mrs.fr

March 11, 2014

Abstract

Given a network and a set of source destination pairs (connections), we consider the problem of maximizing the sum of the flow under proportional delay constraints. In this paper, the delay for crossing a link is proportional to the total flow crossing this link. If a connection supports non-zero flow, then the sum of the delay along any path corresponding to that connection must be lower than a given bound. The constraints of delay are on-off constraints because if a connection does not support non-zero flow, then there is no constraint for that connection. The difficulty of the problem comes from the choice of the connections supporting non-zero flow.

We first prove a general approximation ratio using linear programming for a variant of the problem. We also prove a linear time 2-approximation algorithm when the network is a path. We then show a Polynomial Time Approximation Scheme when the graph of intersection of the paths has bounded treewidth. We finally prove that the problem is NP-complete even when the network is a tree.

Keywords: Maximum flow, on-off delay constraints, polynomial time approximation scheme (PTAS), dynamic programming, bounded treewidth, linear programming, NP-completeness.

1 Introduction

The multi-commodity flow problem is a classical network flow problem with multiple commodities (flow demands) between different source and sink nodes. This problem has been widely studied in the literature. Given a network, a set of capacities on edges, and a set of demands (commodities), the problem consists in finding a network flow satisfying all the demands and respecting capacities

and flow conservation constraints. The integer version of the problem is NP-complete [6], even for only two commodities and unit capacities (making the problem Strongly NP-complete in this case). In that version, the problem consists in producing an integer flow satisfying all demands and respecting the previously mentioned constraints. However, if fractional flows are allowed, the problem can be solved in polynomial time through linear programming [4, 1, 9].

Network operators must satisfy some Quality of Service requirements for their clients. One of the most important parameters in telecommunications networks is the end to end delay of a unit of flow between a source node and a destination node. This requirement is not taken into consideration in the multi-commodity flow problem. The delay through a link depends on the amount of flow supported by this link; classicaly it is modeled by a convex function. The end to end delay for a demand and a path associated with this demand, is the sum of the delay through all links of this path. Some papers focused on minimizing the mean end to end delay. This problem consists in minimizing a convex function under linear constraints [3, 10] and can be solved using semidefinite programming [11]. Other papers focused on finding a multi-commodity flow that satisfies the demands and that minimizes the maximum end to end delay [5].

As the authors of [2], we think that a more realistic problem consists in adding strict end to end delay constraints for all connections. Indeed, in communication networks, there are multiple classes of services, and for each of them, it is crucial to respect a certain level of Quality of Service, that is respecting a threshold for the end to end delay. It is why we study a multicommodity flow problem in which all demands have to respect an end to end delay constraint.

In this paper, we focus on multicommodity network flow problems in which each edge possesses a proportional latency function that describes the common delay experienced by the flow on that edge as a function of the flow rate. We investigate the problem of maximizing the sum of the flow under proportional delay constraints. We allow fractional flows for all connections. As mentioned before, we have a end to end delay constraint for each demand. But, if a path associated with one connection does not support non-zero flow, then the constraint is not active. These on-off constraints make the problem more difficult to solve than just a simple linear program [8].

We formally present our problem in Section 1.1. We then describe a simple example in Section 1.2 and preliminary results in Section 1.3. We present our contributions in Section 1.4.

1.1 Problem formulation and model

Let $G = (V, E)$ be a connected graph (that represents a network) with a coefficient α_e for each edge $e \in E$. Let $\{(s_1, t_1), \ldots, (s_m, t_m)\}$ be a set of m source destination pairs (connections). Let $\mathcal{P} = \{P_1, \ldots, P_m\}$ be a set of m paths in G. The path P_i corresponds to the source destination pair (s_i, t_i) for all $i = 1, \ldots, m$. Without loss of generality, we suppose that there is a unique path for each source destination pair. As described later, the objective is to maximize the sum of the flows, and so if there are several paths for a connection, we assign a source destination pair to each path. In that case (s_i, t_i) can be equal to (s_j, t_j) with $i \neq j$. We denote by x_i the flow through the path P_i for all $i = 1, \ldots, m$. We suppose that the delay τ_e for crossing an edge $e \in E$ is proportional to the total flow $\sum_{i:e\in E(P_i)} x_i$ crossing the edge e, i.e. $\tau_e = \alpha_e \sum_{i:e\in E(P_i)} x_i$. Let $\lambda > 0$. For all $i = 1, \ldots, m$, we require for path P_i that, if $x_i > 0$, then the end to end delay $\sum_{e\in P_i} \tau_e$ is at most λ. By scaling the coefficients of the α_e, we can make this bound λ equal to one. A multicommodity flow $x = (x_1, \ldots, x_m)$ satisfies the latency requirement if for all $j = 1, \ldots, m$ such that $x_j > 0$, the following constraint for path P_j is satisfied: $\sum_{e\in E(P_j)} \tau_e \leq 1$. Using the notation $\beta_{i,j} := \sum_{e\in E(P_i)\cap E(P_j)} \alpha_e$, this constraint can be written as follows: $\sum_{i=1}^{m} \beta_{i,j} x_i \leq 1$. The

Maximum Flow under Delay Constraint problem consists in finding among the solutions satisfying these constraints a solution of maximum value $\sum_{i=1}^{m} x_i$. In other words, the problem consists in solving:

$$\begin{cases} \text{Max} & \sum_{i=1}^{m} x_i \\ & \sum_{i=1}^{m} \beta_{i,j} x_i \leq 1 \quad j = 1, \ldots, m \quad x_j > 0 \\ & x_i \geq 0 \qquad\qquad\quad i = 1, \ldots, m \end{cases}$$

The hardness of this problem comes from the choice of the paths supporting a non-zero flow. Indeed, if we are given a set of paths $\mathcal{P}^* \subseteq \mathcal{P}$ carrying non-zero flow in some optimal solution, then the problem becomes polynomial since it reduces to solve the linear program $LP(\mathcal{P}^*)$. Without loss of generality let $\mathcal{P}^* = \{P_1, \ldots, P_{m^*}\}$ with $m^* \leq m$.

$$LP(\mathcal{P}^*) \begin{cases} \text{Max} & \sum_{i=1}^{m^*} x_i \\ & \sum_{i=1}^{m^*} \beta_{i,j} x_i \leq 1 \quad j = 1, \ldots, m^* \\ & x_i \geq 0 \qquad\qquad\quad i = 1, \ldots, m^* \end{cases}$$

The dual of this linear program is:

$$DLP(\mathcal{P}^*) \begin{cases} \text{Min} & \sum_{j=1}^{m^*} y_j \\ & \sum_{j=1}^{m^*} \beta_{i,j} y_j \geq 1 \quad i = 1, \ldots, m^* \\ & y_j \geq 0 \qquad\qquad\quad j = 1, \ldots, m^* \end{cases}$$

Note that $LP(\mathcal{P}^*)$ and its dual $DLP(\mathcal{P}^*)$ differ only by the sense of inequalities and the direction of optimization. In particular, if the system of equations $\sum_{i=1}^{m^*} \beta_{i,j} x_i = 1, j = 1, \ldots, m^*$, has a solution then this solution is optimal for the primal and for the dual since it satisfies both primal and dual complementary slackness conditions.

In this paper, we do not consider capacity constraints except for the complexity result of the last section. In other words, we can consider that capacities on edges (maximum volume of flows that can be supported by the edges) are sufficiently large, and so we can ignore them. However, the Polynomial Time Approximation Scheme proposed in Section 4 is still correct when adding any capacity constraints and with any end to end delay constraints for connections.

1.2 Example

Consider the path $G = (V, E)$ described in Figure 1 and the three source destination pairs (s_1, t_1), (s_2, t_2), and (s_3, t_3). The path P_i for (s_i, t_i) is the unique simple path between s_i and t_i in G, for all $i \in \{1, 2, 3\}$. We set $\alpha_e = 1$ for all $e \in E$. The Maximum Flow under Delay Constraint problem consists in solving:

$$\begin{cases} \text{Max} & x_1 + x_2 + x_3 \\ & x_1(3x_1 + 2x_2) \leq x_1 \\ & x_2(2x_1 + 4x_2 + x_3) \leq x_2 \\ & x_3(x_2 + 2x_3) \leq x_3 \\ & x_1, x_2, x_3 \geq 0 \end{cases}$$

The first constraint means that, if the path P_1 supports a non-zero flow x_1, then the end to end delay for connection (s_1, t_1) must be at most one. The delay for crossing the leftmost edge of P_1 is x_1. The delay is $x_1 + x_2$ for each of the two other edges. Thus, if $x_1 > 0$, then $3x_1 + 2x_2 \leq 1$.

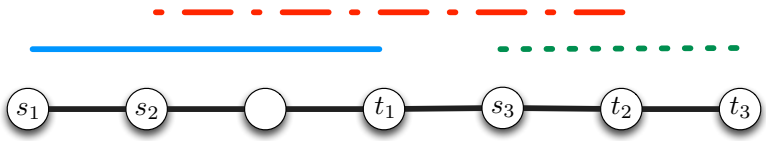

Figure 1: Instance of the Maximum Flow under Delay Constraint problem. The graph $G = (V, E)$ is a path composed of seven nodes and $\alpha_e = 1$ for all $e \in E$. There are three source destination pairs: (s_1, t_1), (s_2, t_2), and (s_3, t_3). The three paths are represented above G.

Note that if $x_1 = 0$, the first constraint is always satisfied. The second and the third inequalities are constructed similarly.

One can observe that $x^* = (1/3, 0, 1/2)$ is the optimal solution. Note that connection (s_2, t_2) does not support flow, that is $x_2 = 0$, and so the second constraint is satisfied. That explains why the end to end delay $2x_1 + 4x_2 + x_3 = 7/6 > 1$ for connection (s_2, t_2) is not satisfied.

1.3 Preliminaries

We first prove in Lemma 1 that we can consider instances such that the path of any connection is not included in a path of another connection (but the paths may intersect).

Lemma 1 *Let I be an instance of the Maximum Flow under Delay Constraint problem. If there are two different connections P_j and P_k such that $E(P_k) \subseteq E(P_j)$, then there is an optimal solution x^* such that $x^*_j = 0$.*

Proof. Consider any admissible flow x for I such that $x_j > 0$. We construct another admissible flow x' from x in which we only change the amount of flow for paths P_j and P_k. More precisely, we set $x'_k := x_k + x_j$, $x'_j := 0$, and $x'_i := x_i$ for all $i \in \{1, \ldots, m\} \setminus \{j, k\}$. Clearly, the total amount of flow is unchanged, that is $\sum_{i=1}^{m} x_i = \sum_{i=1}^{m} x'_i$. Let τ and τ' denote the vector of flow per edge for x and x', respectively. By construction of x', we get $\tau'_e \leq \tau_e$ for all $e \in E$ because $E(P_k) \subseteq E(P_j)$. As x is an admissible flow, then it follows that $\sum_{e \in E(P_i)} \tau'_e \leq 1$, for all $i = 1, \ldots, m$. Thus, the vector of flow x' is admissible. To conclude, if x is an optimal solution, then x' is an optimal solution such that $x'_j = 0$. □

By Lemma 1, we only consider instances such that $E(P_k) \not\subseteq E(P_j)$ for all $j, k \in \{1, \ldots, m\}$, $j \neq k$.

Let us define the graph H of intersection of the paths of an instance of the Maximum Flow under Delay Constraint problem. The set of nodes $V(H) = \{h_1, \ldots, h_m\}$ corresponds to the set of paths $\mathcal{P} = \{P_1, \ldots, P_m\}$. For $i, j \in \{1, \ldots, m\}$, $i \neq j$, there is an edge $\{h_i, h_j\} \in E(H)$ between two nodes $h_i \in V(H)$ and $h_j \in V(H)$ if, and only if, there exists $e \in E$ such that $e \in E(P_i) \cap E(P_j)$, that is when P_i and P_j share at least one edge. The graph of intersection of the paths H of the instance depicted in Figure 1 is a path composed of three nodes. Without loss of generality, we assume that the graph H is a connected graph.

1.4 Contributions

- In Section 2, we analyze the quality of the solution given by the linear program for a variant of the Maximum Flow under Delay Constraint problem in which all the delay constraints must be satisfied even for connections that do not support flow. We prove a general bound for the ratio between the value of such a solution and the value of an optimal solution, and so we get a polynomial approximation algorithm. We deduce a polynomial L-approximation algorithm where L is the size of a longest path of $\mathcal{P}$, and a polynomial 2-approximation algorithm when the graph G is a path.

- In Section 3, we prove a linear time 2-approximation algorithm when G is a path by computing a best independent solution (the paths of connections that support non-zero flow are edge-disjoint). This result asks some interesting open questions about better approximation factors for the Maximum Flow under Delay Constraint problem.

- In Section 4, we prove a Polynomial Time Approximation Scheme when the graph of intersection of the paths H has bounded treewidth. To do that, we first developp an exact dynamic programming algorithm for a variant of the problem in which the possible amount of flow per connection is taken from a given set X. The Polynomial Time Approximation Scheme is based on the fact that the dynamic programming algorithm is polynomial when H has bounded treewidth and when X is a constant (that depends on the approximation ratio). These results hold when adding any capacity constraints and with any end to end delay constraints for connections: for all $i = 1, \ldots, m$, if $x_i > 0$, then the end to end delay $\sum_{e \in P_i} \tau_e$ is at most λ_i.

- In Section 5, we show that the Maximum Flow under Delay Constraint problem is NP-complete even if the graph G is a tree. In our proof, we consider capacity constraints. In [2], it is proved that the problem is NP-complete without capacity constraints but the graphs must contain cycles.

2 Approximation algorithms using linear programming

We first define a variant of the Maximum Flow under Delay Constraint problem, called Maximum Flow under Strong Delay Constraint problem, for which all the end to end delay constraints must be satisfied even for connections that do not support flow. This problem is polynomial since it reduces to solve the following linear program:

$$LP(\mathcal{P}) \begin{cases} \text{Max} & \sum_{i=1}^{m} x_i & \\ & \sum_{i=1}^{m} \beta_{i,j} x_i \leq 1 & j = 1, \ldots, m \\ & x_i \geq 0 & i = 1, \ldots, m \end{cases}$$

Consider the example of Figure 1. The Maximum Flow under Strong Delay Constraint problem consists in solving:

$$\begin{cases} \text{Max} & x_1 + x_2 + x_3 \\ & 3x_1 + 2x_2 \leq 1 \\ & 2x_1 + 4x_2 + x_3 \leq 1 \\ & x_2 + 2x_3 \leq 1 \\ & x_1, x_2, x_3 \geq 0 \end{cases}$$

One can observe that $x = (1/4, 0, 1/2)$ is an optimal solution for this variant of the problem. Such a solution for the Maximum Flow under Strong Delay Constraint problem obtained by solving $LP(\mathcal{P})$ is closed to the optimal solution x^* for the Maximum Flow under Delay Constraint problem. Indeed, $(x_1^* + x_2^* + x_3^*)/(x_1 + x_2 + x_3) = 10/9$.

In this section, we analyze the ratio between the value of an optimal solution for the Maximum Flow under Strong Delay Constraint problem and the value of an optimal solution for the Maximum Flow under Delay Constraint problem. We then deduce some polynomial approximation algorithms based on the resolution of the linear program $LP(\mathcal{P})$ for the Maximum Flow under Strong Delay Constraint problem. More precisely, we first prove a general polynomial approximation algorithm (Theorem 1). We then deduce a polynomial L-approximation algorithm where $L = \max_{i=1,\dots,m} |E(P_i)|$ is the size of a longest path (Corollary 1) and we finally prove a polynomial 2-approximation algorithm when the graph is a path (Corollary 2).

We prove in Theorem 1 a polynomial approximation algorithm based on the resolution of the linear program $LP(\mathcal{P})$ for the Maximum Flow under Strong Delay Constraint problem. Let us first define some parameters and prove Lemma 2.

Definition 1 *For all $k = 1, \dots, m$, let $S_k^* \subseteq \{1, \dots, m\} \setminus \{k\}$ be a maximum cardinality set of indices such that for all $i \in S_k^*$, there exists $e \in E(P_k) \cap E(P_i)$ such that for all $j \in S_k^* \setminus \{i\}$, $e \notin E(P_j)$. Let $|S^*| = \max_{i=1,\dots,m} |S_i^*|$.*

Lemma 2 *Let I be any instance. Let x^* be an optimal solution for the Maximum Flow under Delay Constraint problem and let x be an optimal solution for the Maximum Flow under Strong Delay Constraint problem. Then $\sum_{i=1}^m x_i \geq \sum_{i=1}^m x_i^*/|S^*|$.*

Proof. Recall that if $x_k^* > 0$, then $\sum_{e \in E(P_k)} \tau_e^* \leq 1$. We first prove that if $x_k^* = 0$, then $\sum_{e \in E(P_k)} \tau_e^* \leq |S_k^*|$. Let P_k be a path such that $x_k^* = 0$. Consider a minimal (by inclusion) set of paths $\{P_i : i \in S\}$ supporting non-zero flow that covers all edges of P_k supporting non-zero flow. By definition of S_k^*, $|S| \leq |S_k^*|$. Assign to each edge $e \in E(P_k)$ supporting non-zero flow, a path P_i, $i \in S$ such that $e \in P_i$. For each $i \in S$, let E_i be the set of edges assigned to the path P_i. Since $x_i^* > 0$, $\sum_{e \in E_i} \tau_e^* \leq 1$, that is the sum of the delays on the edges of E_i is at most 1. There are $|S|$ groups of edges, and thus $\sum_{e \in E(P_k)} \tau_e^* \leq |S| \leq |S_k^*|$, that is the total delays on the edges of P_k is at most $|S_k^*|$.

For all $k \in \{1, \dots, m\}$ such that $x_k^* = 0$, then $\sum_{e \in E(P_k)} \tau_e^* \leq |S_k^*|$. Consider the solution x' obtained by dividing all the flows of x^* by $|S^*|$. Formally, set $x_i' := x_i^*/|S^*|$ for all $i = 1, \dots, m$. Recall that $|S^*| = \max_{i=1,\dots,m} |S_k^*|$. By previous claims, x' is an admissible solution. By construction, $\sum_{i=1}^m x_i' = \sum_{i=1}^m x_i^*/|S^*|$. To conclude, one can observe that an optimal solution x of the Maximum Flow under Strong Delay Constraint problem obtained by solving the linear program $LP(\mathcal{P})$ is such that $\sum_{i=1}^m x_i \geq \sum_{i=1}^m x_i'$. Indeed all the end to end delay constraints for x' are satisfied even for connections that do not support flow, and so x' is an admissible solution for the Maximum Flow under Strong Delay Constraint problem. □

By Lemma 2, we deduce a polynomial $|S^*|$-approximation algorithm.

Theorem 1 *There exists a polynomial $|S^*|$-approximation algorithm for the Maximum Flow under Delay Constraint problem.*

Corollary 1 *Let I be any instance and let $L = \max_{i=1,\dots,m} |E(P_i)|$. Let x^* be an optimal solution for the Maximum Flow under Delay Constraint problem and let x be an optimal solution for the*

Maximum Flow under Strong Delay Constraint problem. Then $\sum_{i=1}^{m} x_i \geq \sum_{i=1}^{m} x_i^*/L$. *There exists a polynomial L-approximation algorithm for the Maximum Flow under Delay Constraint problem.*

When the graph G is a path, one can observe that $|S^*| \leq 2$. We deduce in Corollary 2 a polynomial 2-approximation algorithm for such class of instances.

Corollary 2 *Let I be any instance and suppose that G is a path. Let x^* be an optimal solution for the Maximum Flow under Delay Constraint problem and let x be an optimal solution for the Maximum Flow under Strong Delay Constraint problem. Then* $\sum_{i=1}^{m} x_i \geq \sum_{i=1}^{m} x_i^*/2$. *There exists a polynomial 2-approximation algorithm for the Maximum Flow under Delay Constraint problem when the graph is a path.*

Note that we cannot find a similar result when the graph G is a tree. Indeed, $|S^*|$ may be arbitrarily large for this class of graphs.

3 Linear time 2-approximation algorithm when G is a path

This section is devoted to prove another polynomial 2-approximation algorithm when the graph is a path. The interest of this algorithm is its linear time complexity. Furthermore, we conjecture that a generalization could give a polynomial time $(k+1)/k$-approximation algorithm (with $k \geq 1$ a constant integer). A solution x of the Maximum Flow under Delay Constraint problem is said to be *independent* if its support $\{P_k : x_k > 0\}$ is a family of edge-disjoint paths. In other words, x is independent if the set of vertices $\{h_k : x_k > 0\} \subset V(H)$ forms an independent set of H. Recall that the graph H is the graph of intersection of the paths. We prove in Lemma 3 that a best independent solution has a value which is at least half the total flow of an optimal solution. Note that $x = (1/3, 0, 1/2)$ is the best independent solution for the instance depicted in Figure 1 (that coincides in that case to the optimal solution of the Maximum Flow under Delay Constraint problem). In the following, we consider that H is node-weighted. For all $i = 1, \ldots, m$, the weight w_{h_i} (or simply denoted w_i) of the node $h_i \in V(H)$ represents the maximum amount of flow that can support the path P_i if the paths sharing an edge with P_i do not carry any flow. Formally, $w_i := 1/\sum_{e \in E(P_i)} \alpha_e$.

Lemma 3 *Let I be any instance and suppose that G is a path. Let x^* be an optimal solution and let x be a best indepedent solution for the Maximum Flow under Delay Constraint problem. Then* $\sum_{i=1}^{m} x_i \geq \frac{1}{2} \sum_{i=1}^{m} x_i^*$. *There exists a linear time 2-approximation algorithm for the Maximum Flow under Delay Constraint problem when the graph is a path.*

Proof. Consider an optimal solution x^* of an instance I and the instance I' obtained from I by removing all source-destination pairs that do not belong to the support of x^*. Since instances I and I' have the same optimum and an independent solution for I' is also an independent solution for I, it suffices to prove the claim of lemma 3 for I'. For the instance I', there is an optimal solution in which all delay constraints are active. This solution is therefore an optimal solution of the linear program $LP(\mathcal{P}^*)$ where $\mathcal{P}^* := \{P_i : x_i^* > 0\}$ and its value is at most the value of any feasible dual solution y of $DLP(\mathcal{P}^*)$. Without loss of generality let $\mathcal{P}^* = \{P_1, \ldots, P_{m^*}\}$ with $m^* \leq m$. Let $x_{I'}$ be an optimal independent solution of I', we will show that there exists a feasible solution $y_{I'}$ of $DLP(\mathcal{P}^*)$ whose value is at most twice the value of $x_{I'}$.

Since G is a path, the graph of intersection of the paths H is an interval graph. Let $\mathcal{C}$ be the set of maximal cliques of H. Since H is perfect, the caracteristic vector z of maximum weighted

independent set is an optimal solution of the following linear program $LPQ(\mathcal{P}^*)$:

$$(LPQ(\mathcal{P}^*)) \begin{cases} \text{Max} & \sum_{i=1}^{m^*} w_i z_i \\ & \sum_{i:h_i \in C} z_i \leq 1 \quad \forall C \in \mathcal{C} \\ & z_i \geq 0 \qquad\qquad i = 1, \ldots, m^*. \end{cases}$$

Consider the dual $(DLPQ(\mathcal{P}^*))$ of this linear program :

$$(DLPQ(\mathcal{P}^*)) \begin{cases} \text{Min} & \sum_{C \in \mathcal{C}} t_C \\ & \sum_{C:h_i \in C} t_C \geq w_i \quad i = 1, \ldots, m^* \\ & t_C \geq 0 \qquad\qquad \forall C \in \mathcal{C}. \end{cases}$$

Let t be an optimal solution of $DLPQ(\mathcal{P}^*)$. The following algorithm computes from t, a feasible solution y of $DLP(\mathcal{P}^*)$ whose value is at most twice the value of t. Let $C \in \mathcal{C}$, we denote $l(C)$ (resp. $r(C)$) the index of the leftmost (resp. rightmost) path such that its corresponding node in H belongs to the clique C.

1. For all $i = 1, \ldots, m^*$ do
2. $\quad y_i := 0$
3. For all $C \in \mathcal{C}$ do
4. $\quad y_{l(C)} := y_{l(C)} + t_C$
5. $\quad y_{r(C)} := y_{r(C)} + t_C$

For all $i = 1, \ldots, m^*$, if $e \in E(P_i)$ the following inequality holds:

$$\sum_{j:e \in E(P_j)} y_j \geq \sum_{C:h_i \in C} t_C \geq w_i.$$

Since t is a feasible solution of $DLPQ(\mathcal{P}^*)$, the second inequality is trivially true. In order to verify the first inequality, note that if the node h_i corresponding to the path P_i belongs to the clique C then $e \in E(P_i)$ implies that $e \in E(P_{l(C)})$ or $e \in E(P_{r(C)})$. Therefore, if a clique C contributes t_C for the second sum then $y_{l(C)}$ or $y_{r(C)}$ contribubes t_C for the first one. We conclude that, for all paths $i = 1, \ldots, m^*$,

$$\sum_{e \in E(P_i)} \alpha_e \Big(\sum_{j:e \in E(P_j)} y_j \Big) \geq \sum_{e \in E(P_i)} \alpha_e w_i = w_i \sum_{e \in E(P_i)} \alpha_e = 1.$$

This shows that y is a feasible solution of $DLP(\mathcal{P}^*)$. By the definition of y, its value is at most twice the weight of a maximum weight stable set S of H. The independent solution corresponding to S is thus a 2-approximation for the instance I'.

To conclude the proof, computing an optimal independent solution when G is a path reduces to compute a maximum weighted independent set of H and so we get the linear time complexity because H is an interval graph. □

Note that we cannot find a similar result when the graph G is a tree. Indeed, a best independent solution may be arbitrarily far than an optimal solution for this class of graphs.

As mentioned before, we conjecture better approximations for paths generalizing this technique. Let $K \geq 1$ be any constant integer. We construct the graph H^K as follows. The set of vertices of

H^K are all the subsets of paths of size at most K such that any two paths of such a subset share at least one edge. There is an edge between two nodes $u \in V(H^K)$ and $v \in V(H^K)$ if, and only if, there is at least one edge that belongs to one path of the set of u and that belongs to one path of the set of v. The weight of a node u is the maximum amount of flow that can be sent through the paths that belongs to the set of u if the paths (not in the set of u) sharing an edge with at least one path of the set of u do not carry any flow. The graph H^K and the set of weights can be computed in polynomial time because K is constant. The idea is then to compute a maximum weighted independent set of H^K. We conjecture that this procedure gives a $((K+1)/K)$-approximation algorithm when the graph G is a path. Note that $K = 1$ corresponds to the polynomial 2-approximation algorithm proved in Lemma 3.

4 Polynomial Time Approximation Scheme for some classes

We first define a variant of the problem, called Maximum Discrete Flow under Delay Constraint problem, as follows. The only change is that the possible amount of flow for all connections is taken among a finite set of non negative values containing zero. More formally, given a finite set X of non negative values containing zero, $x_i \in X$ for all $i = 1, \ldots, m$. In other words, the Maximum Discrete Flow under Delay Constraint problem consists in solving:

$$\begin{cases} \text{Max} & \sum_{i=1}^{m} x_i & & \\ & \sum_{i=1}^{m} \beta_{i,j} x_i \leq 1 & j = 1, \ldots, m & x_j > 0 \\ & x_i \in X & i = 1, \ldots, m & \end{cases}$$

Consider the instance described in Figure 1 with $X = \{0, 1/3, 2/3, 1\}$. The Maximum Discrete Flow under Delay Constraint problem consists in solving:

$$\begin{cases} \text{Max} & x_1 + x_2 + x_3 \\ & x_1(3x_1 + 2x_2) \leq x_1 \\ & x_2(2x_1 + 4x_2 + x_3) \leq x_2 \\ & x_3(x_2 + 2x_3) \leq x_3 \\ & x_1, x_2, x_3 \in \{0, 1/3, 2/3, 1\} \end{cases}$$

One can observe that $x = (1/3, 0, 1/3)$ is an optimal solution for this variant of the problem.

In this section, we prove an exact dynamic programming algorithm for the Maximum Discrete Flow under Delay Constraint problem based on a tree decomposition of H (Lemma 4). We then deduce a Polynomial Time Approximation Scheme for the Maximum Flow under Delay Constraint problem when the graph of intersection of the paths H has bounded treewidth (Theorem 2).

Lemma 4 *Let X be a finite set of non negative values containing zero. There exists an exact $O(m|X|^{tw(H)+1})$-time complexity algorithm for the Maximum Discrete Flow under Delay Constraint problem where $tw(H)$ is the treewidth of H.*

Proof. Consider a tree decomposition T of H, with set of nodes $V(T) = \{Y_1, \ldots, Y_n\}$, where each Y_i is a subset of $V(H)$, satisfying the following properties:

- The union $\cup_{i=1}^{n} Y_i$ of all sets Y_i equals $V(H)$, that is each vertex $h \in V(H)$ is contained in at least one node of T.

- If Y_i and Y_j both contain a vertex $h \in V(H)$, then all nodes Y_k of T in the (unique) path between Y_i and Y_j contain h as well. Equivalently, the tree nodes containing vertex h form a connected subtree of T.

- For every edge $\{h, h'\} \in E(H)$, there is a subset Y_i that contains both h and h', that is vertices are adjacent in H only when the corresponding subtrees have a node in common.

We suppose that T is such that $tw(H) = \max_{i=1,\dots,n} |Y_i|-1$. Let $t = tw(H)+1$. Let $r \in V(T)$ be the root (arbitrarily chosen) of the tree T. The set $N(u)$ represents the children of u for all $u \in V(T)$. Let $d_u = |N(u)|$. The subtree T_u is the connected component of T containing u when removing the edge $\{u, u'\}$ where u' is the parent of u, for all $u \in V(T) \setminus \{r\}$. Recall that the set of nodes $V(H) = \{h_1, \dots, h_m\}$ corresponds to the set of source destination pairs, and so corresponds to the set of paths $\mathcal{P} = \{P_1, \dots, P_m\}$. Let us define $P_u = \{P_i : \exists v \in V(T_u) : h_i \in v, h_i \notin u', i = 1, \dots, m\}$ and let $Q_u = \{P_i : \exists v \in V(T_u) : h_i \in v, h_i \in u', i = 1, \dots, m\}$ where u' is the parent of u in T, for all $u \in V(T)$. Note that the set $P_u \cup Q_u$ represents all the paths that correspond to nodes of subtree T_u.

Let $u \in V(T)$. Let $\{q_u^1, q_u^2, \dots, q_u^{|X|^{|Q_u|}}\}$ be the set of all possible vectors of flows for the set of paths Q_u. For all $i = 1, \dots, |X|^{|Q_u|}$, let x_u^i be an optimal solution for the sub-problem induced by the set of paths $P_u \cup Q_u$ when the vector of flows for paths of Q_u is q_u^i. Without loss of generality, we suppose that the vector q_u^i is admissible for all $u \in V(T)$ and for all $i = 1, \dots, |X|^{|Q_u|}$.

We aim at computing vectors x_u^i for all $i = 1, \dots, |X|^{|Q_u|}$ and for all $u \in V(T)$. As $|Q_u| \leq t$, then there are $O(|X|^t)$ vectors to compute for each u.

We proceed by induction. Consider any leaf $u \in V(T)$. Then $|P_u \cup Q_u| \leq t$ by construction of T. Thus, we compute x_u^i for all $i = 1, \dots, |X|^{|Q_u|}$ by enumerating all the possible vectors for paths of Q_u. This can be done in $O(|X|^{|P_u \cup Q_u|}) = O(|X|^t)$-time.

Consider a node $u \in V(T)$ such that u is not a leaf of T. Let $N(u) = \{u_1, \dots, u_{d_u}\}$ be the set of children of u. Suppose we have computed $x_{u_j}^i$ for all $j = 1, \dots, d_u$ and for all $i = 1, \dots, |X|^{|Q_{u_j}|}$. We now compute x_u^i for all $i = 1, \dots, |X|^{|Q_u|}$. In other words, we compute for all possible vectors of flows for paths of Q_u, an optimal solution for the sub-problem induced by the set of paths $Q_u \cup P_u$. To do that, let $R_u = P_u \setminus \cup_{j=1}^{d_u} P_{u_j}$. We compute for all possible vectors of flows for paths of $Q_u \cup R_u$, an optimal solution for the sub-problem induced by the set of paths $Q_u \cup R_u \cup_{j=1}^{d_u} P_{u_j}$. Let $\{r_u^1, \dots, r_u^{|Q_u \cup R_u|}\}$ be the set of all possible vectors of flow for $Q_u \cup R_u$. Note that $|Q_u \cup R_u| \leq t$. By construction of T and by definition of P and Q, if a path $P \in P_{u_{j'}}$, for some $j' \in \{1, \dots, d_u\}$, then $P \notin P_{u_j}$ for all $j \in \{1, \dots, d_u\} \setminus \{j'\}$. Thus, for all $i = 1, \dots, |X|^{|Q_u \cup R_u|}$, we consider the vector of flow r_u^i and we compute an optimal solution for the sub-problem induced by the set of paths $Q_u \cup R_u \cup P_{u_j}$ for all $j = 1, \dots, d_u$. By previous remarks, that can be done independently for each j, and so such a computation can be done in $O(d_u|X|^t)$-time because $|Q_u \cup R_u| \leq t$. Finally, for all possible vectors of flows for paths of Q_u, we keep an optimal solution for the sub-problem induced by the set of paths $Q_u \cup P_u$. This can be done in $O(|X|^t)$-time. Thus, we have computed x_u^i for all $i = 1, \dots, |X|^{|Q_u|}$.

We now prove that, for all $i = 1, \dots, |X|^{|Q_u|}$, x_u^i represents an optimal vector for the sub-problem induced by the set of paths $P_u \cup Q_u$ when the vector of flow for paths of Q_u is q_u^i. Suppose it is not the case and let $i \in \{1, \dots, |X|^{|Q_u|}\}$ be such that $x = x_u^i$ is not an optimal vector. Let x' be an optimal vector for the sub-problem induced by the set of paths $P_u \cup Q_u$. Without loss of generality, the $|P_u|$ first indices correspond to flows for paths of P_u. Furthermore, the $|\cup_{j=1,\dots,d_u} P_{u_j}|$ first indices correspond to flows for paths of $\cup_{j=1,\dots,d_u} P_{u_j}$. By definition, $x_k = x'_k$

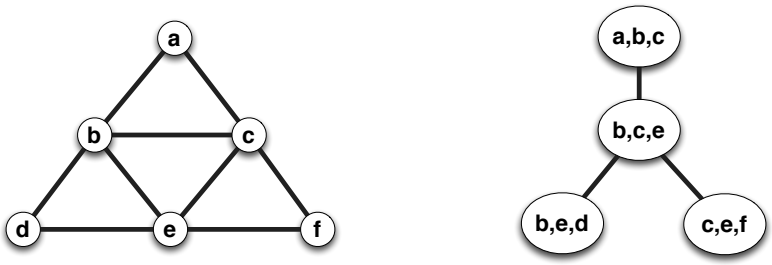

Figure 2: A graph of intersection of the paths H (left) and an optimal tree decomposition T of H with $tw(T) = 2$ (right).

for all $k \in \{|P_u| + 1, \ldots, |P_u| + |Q_u|\}$. Furthermore, as we have computed x_u among all possible vectors of flows for paths of Q_{u_j} for all $j \in \{1, \ldots, d_u\}$, then we assume that $x_k = x'_k$ for all $k \in \{|\cup_{j=1,\ldots,d_u} P_{u_j}| + 1, \ldots, |P_u|\}$. Thus, there exists $j \in \{1, \ldots, d_u\}$ such that the sum of the flows for the set of paths $P_{u_j} \cup Q_{u_j}$ with x' is strictly greater than for x. As $P_{u_{j_1}} \cap P_{u_{j_2}} = \emptyset$ for all $j_1, j_2 \in \{1, \ldots, d_u\}$, $j_1 \neq j_2$, then it means that there exists $k \in \{1, \ldots, |X|^{Q_{u_j}}\}$ such that $x^k_{u_j}$ is not optimal for the sub-problem induced by the set of paths $P_{u_j} \cup Q_{u_j}$ when the vector of flow for the paths of Q_{u_j} is $q^k_{u_j}$. A contradiction with the induction hypothesis. Thus, for all $i = 1, \ldots, |X|^{|Q_u|}$, x^i_u represents an optimal vector for the sub-problem induced by the set of paths $P_u \cup Q_u$ when the vector of flow for paths of Q_u is q^i_u.

Finally, we have computed x^i_u for all $i = 1, \ldots, |X|^{|Q_u|}$ and for all $u \in V(T)$. As $Q_r = \emptyset$, we deduce an optimal vector of flow choosing the unique vector x^1_r of the root of T. Thus $x^* = x^1_r$ is an optimal solution for the Maximum Discrete Flow under Delay Constraint problem. The complexity of the algorithm is $O(|X|^t \sum_{u \in V(T)} d_u) = O(|E(T)||X|^t) = O(m|X|^{tw(H)+1})$. □

To illustrate the proof of Lemma 4, consider an instance of the Maximum Flow under Delay Constraint problem for which the graph of intersection of the paths H is the graph depicted in Figure 2 (left). The tree T depicted in Figure 2 (right) is an optimal tree decomposition of H with $tw(H) = 2$. Consider any set X of positive values containing 0. We now explain the computation of an optimal solution for the Maximum Discrete Flow under Delay Constraint problem based on the tree decomposition.

1. We first enumerate all the possible admissible vectors of flows for the set of connections $\{b, e, d\}$. There are $O(X^3)$ such vectors.

2. Similarly, we enumerate all the possible admissible vectors of flows for the set of connections $\{c, e, f\}$. There are $O(X^3)$ such vectors.

3. For all possible admissible vectors of flows for $\{b, c, e\}$, we compute the optimal flows for d and f. Since d and f are disjoint connections, we can do that computation sequentially.

   (a) We choose the best value for d among the admissible vectors of flows for $\{b, e, d\}$ computed in 1 by considering only vectors that correspond to the fixed flows for b and e. Indeed, the paths of c and f do not intersect the path of connection d.

(b) We choose the best value for f among the admissible vectors of flows for $\{c, e, f\}$ computed in 2 by considering only vectors that correspond to the fixed flows for c and e. Indeed, this computation is independent of the choices for b and d.

4. For all possible admissible vectors of flows for $\{a, b, c\}$, we compute the optimal flows for d, f, and e using the computations that have been done in 3. Indeed, in 3, we have computed for all the possible vectors of flows for $\{b, c, e\}$, the optimal flows for d and f.

5. We finally compute an optimal solution for the Maximum Discrete Flow under Delay Constraint problem by chosing an optimal vector among the set of vectors computed in 4.

Let $x_{max} = \max_{i=1,\ldots,m} 1/\sum_{e\in E(P_i)} \alpha_e$ and $x_{min} = \min_{i=1,\ldots,m} 1/\sum_{e\in E(P_i)} \alpha_e$. We prove in Theorem 2 a Polynomial Time Approximation Scheme for the Maximum Flow under Delay Constraint problem when the graph H has bounded treewidth and x_{max}/x_{min} is bounded by a constant.

Theorem 2 *Let $b, t \geq 1$ be any constant values. For any $\varepsilon > 0$, there is a polynomial $(1+\varepsilon)$-approximation algorithm for the Maximum Flow under Delay Constraint problem when $tw(H) \leq t$ and $x_{max}/x_{min} \leq b$.*

Proof. To prove Theorem 2, we show that any optimal solution x^ε for an instance of the Maximum Discrete Flow under Delay Constraint problem (the set X will be described later) is such that $\sum_{i=1}^m x_i^\varepsilon \geq (\sum_{i=1}^m x_i^*)/(1+\varepsilon)$, where x^* is an optimal solution for the Maximum Flow under Delay Constraint problem when $x_{max}/x_{min} \leq b$ and $tw(H) \leq t$.

As $tw(H) \leq t$, then there exists an independent set $IS(H)$ of H such that $|IS(H)| \geq |V(H)|/(t+1)$. Indeed H has degeneracy at most t, and then we construct $IS(H)$ as follows: we add one node of degree at most t in $IS(H)$, we remove the neigbhors of this node, and we repeat this process on the remaining graph (of degeneracy at most t). The number of steps is at least $V(H)/(t+1)$. Let R be any constant such that $R \geq m/|IS(H)|$. Recall that $m = V(H)$.

Let ε' be such that $0 < \varepsilon' < \varepsilon$. Let $X_0 = \{x_{max}/(1+\varepsilon')^i : i = 0, 1, \ldots, p\}$ where p is such that $x_{max}/(1+\varepsilon')^p \leq (\varepsilon-\varepsilon')x_{min}/R$ and $X = \{0\}\cup X_0 \cup \{x_{min}\}$. Thus, $(1+\varepsilon')^p \geq x_{max}R/((\varepsilon-\varepsilon')x_{min})$, and so $p \geq \log_{1+\varepsilon'}(x_{max}R/((\varepsilon-\varepsilon')x_{min}))$. One can check that p is constant because $x_{max}/x_{min} \leq b$, with b a constant, R is also a constant, and ε' only depends on ε (*e.g.* $\varepsilon' = \varepsilon/2$). Thus, $|X| = O(1)$.

By Lemma 4, we compute in polynomial time an optimal solution x^ε for the Maximum Discrete Flow under Delay Constraint problem with X as set of possible flow values. Recall that t and $|X|$ are constant.

Let us define $f^+(x) = \sum_{i=1}^m x_i \mathbb{1}_{x_i \geq x_{max}/(1+\varepsilon')^p}$ and $f^-(x) = \sum_{i=1}^m x_i \mathbb{1}_{x_i < x_{max}/(1+\varepsilon')^p}$ for any vector x. We set $f(x) = f^+(x) + f^-(x)$. Consider an optimal solution x^* for the Maximum Flow under Delay Constraint problem. We get $f(x^*) < f^+(x^*) + mx_{max}/(1+\varepsilon')^p$.

We construct an auxiliary vector x' from x^* as follows. For all $i = 1, \ldots, m$, if there exists j, $1 \leq j \leq p$, such that $x_{max}/(1+\varepsilon')^j \leq x_i^* \leq x_{max}/(1+\varepsilon')^{j-1}$, then set $x'_i = x_{max}/(1+\varepsilon')^j$. For all $i = 1, \ldots, m$, if $x_i^* < x_{max}/(1+\varepsilon')^p$, then set $x'_i = 0$. We get $f^+(x^*)/f(x') \leq 1+\varepsilon'$ by construction of x'. As x' takes flow values in X, then $f(x^\varepsilon) \geq f(x')$ because x^ε is an optimal solution for the Maximum Discrete Flow under Delay Constraint problem. Thus, $f^+(x^*)/f(x^\varepsilon) \leq 1+\varepsilon'$.

We now prove that $f(x^*)/f(x^\varepsilon) \leq 1+\varepsilon$. By previous claims, it is sufficient to prove that $f^+(x^*)/f(x^\varepsilon) + (mx_{max})/((1+\varepsilon')^p f(x^\varepsilon)) \leq 1+\varepsilon$. First, recall that $f^+(x^*)/f(x^\varepsilon) \leq 1+\varepsilon'$. It remains to prove that $(x_{max}m)/((1+\varepsilon')^p f(x^\varepsilon)) \leq \varepsilon - \varepsilon'$.

As $R \geq m/|IS(H)|$ and $x_{min} \in X$, then $f(x^\varepsilon) \geq mx_{min}/R$. Indeed the polynomial time algorithm described in Lemma 4 finds, at least, a set of m/R disjoint paths (connections) with

amount of flow (at least) x_{min} (other connections can be zero). The corresponding nodes forms an independent set of H. Recall that p is such that $x_{max}/(1+\varepsilon')^p \leq (\varepsilon - \varepsilon')x_{min}/R$, and so $x_{min} \geq (x_{max}R)/((1+\varepsilon')^p(\varepsilon-\varepsilon'))$. Thus, $f(x^\varepsilon) \geq (x_{max}m)((1+\varepsilon')^p(\varepsilon-\varepsilon'))$, and so $(x_{max}m)/((1+\varepsilon')^d f(x^\varepsilon)) \leq \varepsilon - \varepsilon'$. We conclude that $f(x^*)/f(x^\varepsilon) \leq 1+\varepsilon$, and so the polynomial time algorithm of Lemma 4 is a $(1+\varepsilon)$-approximation algorithm for the Maximum Flow under Delay Constraint problem. □

Let us first define $\chi_e = |\{i : E(P_i) \cap e \neq \emptyset, i = 1, \ldots, m\}|$ for all $e \in E$, as the number of paths that contain edge e. Let $\chi_G = \max_{e \in E} \chi_e$ be the maximum number of paths that share an edge. Let Δ_G be the maximum degree of G. We directly deduce a Polynomial Time Approximation Scheme when the graph G is a bounded degree tree, and x_{max}/x_{min} and χ_G are bounded by constants.

Corollary 3 *Let $b, d, t \geq 1$ be any constant values. For any $\varepsilon > 0$, there is a polynomial $(1+\varepsilon)$-approximation algorithm for the Maximum Flow under Delay Constraint problem when the graph $G = (V, E)$ is a tree, $x_{max}/x_{min} \leq b$, $\Delta_G \leq d$, and $\chi_G \leq t$.*

We also deduce a Polynomial Time Approximation Scheme when the graph G is a path, and x_{max}/x_{min} and χ_G are bounded by constants.

Corollary 4 *Let $b, t \geq 1$ be any constant values. For any $\varepsilon > 0$, there is a polynomial $(1+\varepsilon)$-approximation algorithm for the Maximum Flow under Delay Constraint problem when the graph G is a path, $x_{max}/x_{min} \leq b$, and $\chi_G \leq t$.*

However, the complexity of the Maximum Flow under Delay Constraint problem remains open when the graph G is a path if χ_G is not bounded. More precisely, we do not know if the problem is NP-hard and if there exists a polynomial approximation algorithm with approximation ratio less than 2.

To conclude this section, we observe that the previous results still hold with any capacity constraints and with any end to end delay constraints for connections.

- *Capacity constraints.* For all $e \in E$, we consider the following constraint: $\sum_{i:e \in E(P_i)} x_i \leq c_e$, where c_e represents the maximum amount of flow that can support e.
- *End to end delay constraints.* For all $i = 1, \ldots, m$, we require for path P_i that, if $x_i > 0$, then the end to end delay $\sum_{e \in P_i} \tau_e$ is at most λ_i.

The problem is now:

$$\begin{cases} \text{Max} & \sum_{i=1}^m x_i & & \\ & \sum_{i=1}^m \beta_{i,j} x_i \leq \lambda_j & j = 1, \ldots, m & x_j > 0 \\ & \sum_{i:e \in E(P_i)} x_i \leq c_e & \forall e \in E & \\ & x_i \geq 0 & i = 1, \ldots, m & \end{cases}$$

This new version may change the values of x_{max} and x_{min} but the dynamic programming algorithm of Lemma 4 for the Maximum Discrete Flow under Delay Constraint problem is unchanged. Thus, Lemma 4 and Theorem 2 are still correct when adding the capacity constraints and when replacing 1 by λ_i for all $i = 1, \ldots, m$ assuming that x_{max}/x_{min} is bounded by a constant.

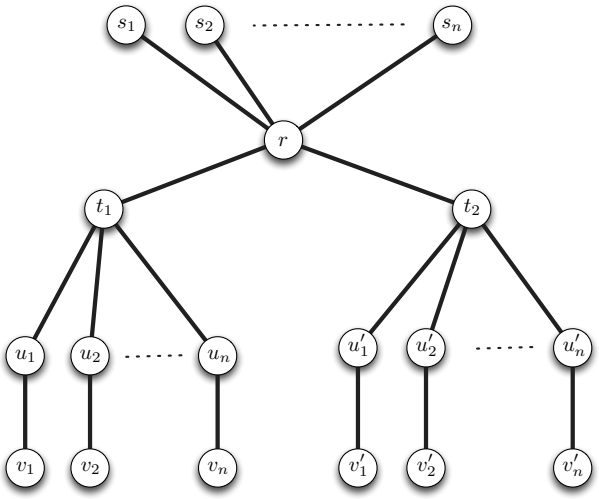

Figure 3: Graph used in the proof of Theorem 3.

5 Hardness result for trees

This section is devoted to prove that the Maximum Flow under Delay Constraint problem is NP-complete even if the graph G is a tree (Theorem 3). In our proof, we consider capacity constraints on edges. Recall that the problem has been proved NP-complete in [2] without capacity constraints for convex delay function but the question has been let open when the graph is a tree. Indeed their proof involves a choice among two different paths for each connection, and so forces the graph G to have cycles.

Theorem 3 *The Maximum Flow under Delay Constraint problem is NP-complete.*

Our reduction uses Partition problem, a well known NP-complete problem [7]. Let $n \geq 2$ and $N = \{1, \ldots, n\}$. Let $I = (p_1, \ldots, p_n)$ such that $\sum_{i=1}^{n} p_i = 2M$. The problem Partition consists in deciding whether there exists a subset $S \subset N$ such that $\sum_{i \in S} p_i = M$.

Let $G = (V, E)$ be a tree with $V = \{s_1, \ldots, s_n\} \cup \{r, t, t'\} \cup \{u_1, \ldots, u_n\} \cup \{u'_1, \ldots, u'_n\} \cup \{v_1, \ldots, v_n\} \cup \{v'_1, \ldots, v'_n\}$ and $E = \{r, t\} \cup \{r, t'\} \cup_{i=1}^{n} \{s_i, r\} \cup_{i=1}^{n} \{t, u_i\} \cup_{i=1}^{n} \{t', u'_i\} \cup_{i=1}^{n} \{u_i, v_i\} \cup_{i=1}^{n} \{u'_i, v'_i\}$. The graph G is depicted in Figure 3. In the following, we abuse the notation writing $\alpha_{u,v}$ instead of $\alpha_{\{u,v\}}$ for all $e \in E$. We denote by $\tau_{u,v}$ the delay for crossing any edge $\{u, v\} \in E$. We set $\alpha_{u,v} = 1$ for all $\{u, v\} \in E$. Thus, $\tau_{u,v}$ represents the amount of flow supported by $\{u, v\}$. We consider the capacity constraints $\tau_{u,v} \leq c_{u,v}$ where $c_{u,v}$ represents the maximum amount of flow that can support any edge $\{u, v\} \in E$. For all $i = 1, \ldots, n$, $c_{s_i,r} = p_i$, and $c_{r,t} = c_{r,t'} = M$. Other capacities are infinite (or sufficiently large). The source destination pairs are (s_i, u_i), (s_i, u'_i), (t, v_i), (t', v'_i) for all $i = 1, \ldots, n$. We denote by $x_{u,v}$ the amount of flow of any connection (u, v). We consider general delay constraints as described at the end of Section 4. We denote by $\lambda_{u,v}$ the maximum end to end delay for any connection (u, v). For all $i = 1, \ldots, n$, $\lambda_{s_i,u_i} = \lambda_{s_i,u'_i} = 2p_i + M$ and $\lambda_{t,v_i} = \lambda_{t',v'_i} = 2\varepsilon$ where $0 < \varepsilon < 1/n$.

Note that $tw(H) \geq n-1 = \Omega(|V(H)|)$. Thus, the Polynomial Time Approximation Scheme described in Section 4 does not work for this class of instances.

Let $P_{u,v}$ be the unique path of the tree G between node $u \in V$ and node $v \in V$. The Maximum Flow under Delay Constraint problem consists in solving:

$$\left\{\begin{array}{lllll} \text{Max} & \sum_{i=1}^{n} x_{s_i,u_i} + x_{s_i,u'_i} + x_{t,v_i} + x_{t',v'_i} & & & \\ & \sum_{e \in P_{s_i,u_i}} \tau_e \leq 2p_i + M & i = 1,\ldots,n & x_{s_i,u_i} > 0 \\ & \sum_{e \in P_{s_i,u'_i}} \tau_e \leq 2p_i + M & i = 1,\ldots,n & x_{s_i,u'_i} > 0 \\ & \sum_{e \in P_{t,v_i}} \tau_e \leq 2\varepsilon & i = 1,\ldots,n & x_{t,v_i} > 0 \\ & \sum_{e \in P_{t',v'_i}} \tau_e \leq 2\varepsilon & i = 1,\ldots,n & x_{t',v'_i} > 0 \\ & \tau_{s_i,r} \leq p_i & i = 1,\ldots,n & \\ & \tau_{r,t}, \tau_{r,t'} \leq M & & \\ & x_{s_i,u_i}, x_{s_i,u'_i}, x_{t,v_i}, x_{t',v'_i} \geq 0 & i = 1,\ldots,n & \end{array}\right.$$

Finally, as the construction of the instance can be done in polynomial time, to prove Theorem 3, it suffices to prove Lemma 5.

Lemma 5 *There exists an admissible vector of flow x such that $\sum_{i=1}^{n} x_{s_i,u_i} + x_{s_i,u'_i} + x_{t,v_i} + x_{t',v'_i} \geq 2M + n\varepsilon$ if, and only if, there exists a solution for the instance I of Partition problem.*

Proof. ($\Leftarrow$). Suppose there exists a solution for the instance I of Partition problem. Let $S \subset N$ be such that $\sum_{i \in S} p_i = M$. Let $S' = N \setminus S$. Note that $\sum_{i \in S'} p_i = M$. We construct the vector of flow x as follows. For all $i \in S$, $x_{s_i,u_i} := p_i$, $x_{s_i,u'_i} := 0$, $x_{t,v_i} := 0$, and $x_{t',v'_i} := \varepsilon$. For all $i \in S'$, $x_{s_i,u'_i} := p_i$ and $x_{s_i,u_i} := 0$, $x_{t',v'_i} := 0$, and $x_{t,v_i} := \varepsilon$. The total amount of flow is $\sum_{i=1}^{n} x_{s_i,u_i} + x_{s_i,u'_i} + x_{t,v_i} + x_{t',v'_i} = 2M + n\varepsilon$.

We now prove that x is an admissible solution. By construction, $\tau_{r,t} = \tau_{r,t'} = M$, and $\tau_{s_i,r} = p_i$ for all $i = 1,\ldots,n$. Thus, the capacity constraints are satisfied. Furthermore, for all $i \in S$, $\tau_{t,u_i} = p_i$, and for all $j \in S'$, $\tau_{t',u'_j} = p_j$. Thus, for all $i = 1,\ldots,n$, the total delay for connection (s_i, u_i) is $2p_i + M$ and so the delay constraints are satisfied. For all $i \in S$, $x_{t,v_i} = 0$ and for all $j \in S'$, $x_{t',v'_j} = 0$, and so the delay constraints are not considered by definition of the problem. For all $i \in S$, $x_{t',v'_i} = \varepsilon$ and for all $j \in S'$, $x_{t,v_j} = \varepsilon$, and so the delay constraints are satisfied because by construction $\tau_{t',u'_i} = \varepsilon$ and $\tau_{t,u_j} = \varepsilon$. Recall that $\lambda_{t',v'_i} = 2\varepsilon$ and $\lambda_{t,v_j} = 2\varepsilon$. Finally, x is an admissible flow such that $\sum_{i=1}^{n} x_{s_i,u_i} + x_{s_i,u'_i} + x_{t,v_i} + x_{t',v'_i} = 2M + n\varepsilon$.

($\Rightarrow$). First, for any vector of flow x, then $\sum_{i=1}^{n} x_{s_i,u_i} + x_{s_i,u'_i} \leq 2M$ because $c_{r,t} = c_{r,t'} = M$.

Suppose there does not exist a solution for the instance I of Partition problem. We prove that for any admissible vector of flow x, then $\sum_{i=1}^{n} x_{s_i,u_i} + x_{s_i,u'_i} + x_{t,v_i} + x_{t',v'_i} < 2M + n\varepsilon$.

First assume that $\sum_{i=1}^{n} x_{s_i,u_i} + x_{s_i,u'_i} = 2M$. Since there does not exist a solution for the instance I of Partition problem, then there exists an $i \in N$ such that $x_{s_i,u_i} \neq 0$ and $x_{s_i,u'_i} \neq 0$. Without loss of generality, assume that $i = 1$ satisfies the previous property. Since $\sum_{i=1}^{n} x_{s_i,u_i} + x_{s_i,u'_i} = 2M$, then $x_{s_1,u_1} + x_{s_1,u'_1} = p_1$. Let $x_{s_1,u_1} = \mu p_1$ and $x_{s_1,u'_1} = (1-\mu)p_1$ with $0 < \mu < 1$. The delay constraint for connection (t, v_1) gives $2x_{t,v_1} + x_{s_1,u_1} \leq 2\varepsilon$. Thus, it means that $x_{t,v_1} \leq \varepsilon - \mu p_1/2$. The delay constraint for connection (t', v'_1) gives $x_{t',v'_1} \leq \varepsilon - (1-\mu)p_1/2$. Thus, $x_{t,v_1} + x_{t',v'_1} \leq 2\varepsilon - p_1/2$. As the p_i's are positive integers and $\varepsilon < 1/n$ with $n \geq 2$, then $p_1/2 > \varepsilon$ and so $x_{t,v_1} + x_{t',v'_1} < \varepsilon$. For all $i = 2,\ldots,n$:

- either $x_{t,v_i} + x_{t',v'_i} < \varepsilon$,

- or $x_{t,v_i} = \varepsilon$ and $x_{t',v'_i} = 0$,
- or $x_{t,v_i} = 0$ and $x_{t',v'_i} = \varepsilon$.

Thus, $\sum_{i=1}^{n} x_{t,v_i} + x_{t',v'_i} < n\varepsilon$, and so $\sum_{i=1}^{n} x_{s_i,u_i} + x_{s_i,u'_i} + x_{t,v_i} + x_{t',v'_i} < 2M + n\varepsilon$.

Assume that $\sum_{i=1}^{n} x_{s_i,u_i} + x_{s_i,u'_i} < 2M$. If $x_{s_i,u_i} + x_{s_i,u'_i} > 0$, for all $i = 1, \ldots, n$, the proof is similar to the previous case replacing p_i by $x_{s_i,u_i} + x_{s_i,u'_i}$. If there exists an $i \in N$ such that $x_{s_i,u_i} + x_{s_i,u'_i} = 0$, then $\sum_{i=1}^{n} x_{s_i,u_i} + x_{s_i,u'_i} \leq 2M - 1$ because the p_i's are positive integers. Since $x_{t,v_i} \leq \varepsilon$ and $x_{t',v'_i} \leq \varepsilon$ for all $i = 1, \ldots, n$, we get $\sum_{i=1}^{n} x_{s_i,u_i} + x_{s_i,u'_i} + x_{t,v_i} + x_{t',v'_i} < 2M + n\varepsilon$ because $\varepsilon < 1/n$. □

6 Future works

We now recall the questions we have asked throughout the paper and add some new questions.

- What is the complexity of the Maximum Flow under Delay Constraint problem when the graph G is a path? A first interesting open question is to determine the complexity of the Maximum Flow under Delay Constraint problem when the graph G is a path such that all (connections) paths share an edge $e \in E$, that is when $\chi_G = m = |\mathcal{P}|$.
- Is there a polynomial approximation algorithm with approximation ratio less than 2 when the graph G is a path with χ_G unbounded? To answer this question, an idea is to analyze the value of an maximum independent set of H^K. What is the approximation ratio obtained by computing a best independent solution for H^K?
- Is there an exact polynomial time algorithm for the Maximum Flow under Delay Constraint problem when $tw(H)$ is bounded?
- Is there classes of instances with $tw(H)$ unbounded that admit a Polynomial Time Approximation Scheme for the Maximum Flow under Delay Constraint problem?
- Is the Maximum Flow under Delay Constraint problem in APX?
- An interesting work is to develop polynomial reduction rules for the Maximum Flow under Delay Constraint problem. For instance, we may use information given by the solutions of the Maximum Flow under Strong Delay Constraint problem as follows. Consider any (connection) path $P_i \in \mathcal{P}$. For which classes of instances can we remove P_i if $LP(\mathcal{P} \setminus \{P_i\}) > LP(\mathcal{P})$? For which classes of instances, $LP(\mathcal{P} \setminus \{P_i\}) < LP(\mathcal{P})$ implies the existence of an optimal solution such that $x_i > 0$?

To conclude, we are investigating a problem in which we are allowed to reduce the delay for a subset of links in respect with a certain given budget. As for the Maximum Flow under Delay Constraint problem, we aim at finding a set of links to modify in order to maximize the sum of the flows respecting the budget given for these transformations.

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[2] Walid Ben-Ameur and Adam Ouorou. Mathematical models of the delay constrained routing problem. *Algorithmic Operations Research*, 1(2), 2006.

[3] Dimitri Bertsekas and Robert Gallager. *Data networks (2nd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[4] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[5] José R. Correa, Andreas S. Schulz, and Nicolás E. Stier Moses. Fast, fair, and efficient flows in networks. *Operations Research*, 55(2):215–225, 2007.

[6] Shimon Even, Alon Itai, and Adi Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5(4):691–703, 1976.

[7] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[8] Hassan Hijazi, Pierre Bonami, Gérard Cornuéjols, and Adam Ouorou. Mixed-integer nonlinear programs featuring "on/off" constraints. *Comp. Opt. and Appl.*, 52(2):537–558, 2012.

[9] Michel Minoux. Networks synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, 19(3):313–360, 1989.

[10] A. Ouorou, P. Mahey, and J.-Ph. Vial. A survey of algorithms for convex multicommodity flow problems, 1997.

[11] C. Touati, E. Altman, and J. Galtier. Semi-definite programming approach for bandwidth allocation and routing in networks. *Game Theory and Applications*, 9:169–179, 2003.