Hardness Results and Heuristic for Multi-groups Interconnection

LÉLIA BLIN¹, CHRISTIAN LAFOREST², STEPHANE ROVEDAKIS¹,* AND NICOLAS THIBAULT³

¹Tour Evry 2, IBISC, Université d’Evry, 523 place des terrasses, 91000 Evry, France
²LIMOS, Complexe scientifique des Cézeaux, 63173 AUBIERE cedex, France
³Paris 2, ERMES CNRS UMR 7181
*Corresponding author: stephane.rovedakis@ibisc.fr

This paper is dedicated to the connection, by a provider, of multiple groups of nodes spread over a network. The role of the provider is to interconnect the members of every group. For this purpose, it must distribute the available links of the network between the groups. The general aim then is to allocate these links in such a way that the communications latencies in the allocated structure are equivalent to the ones in the original (full) network for each group. We study two approaches constructing structures preserving the maximum latency (called the diameter). Unfortunately we show that the associated optimization graph problems are difficult (one cannot be approximated by a constant and the other is NP-complete). Due to these difficulties we relax the constraint on the diameter and propose to construct a unique tree connecting all the groups together. We give a heuristic to treat this problem and we propose several analytical results on its maximum and average latencies performance.

Keywords: network; graph; heuristic; hardness

Received 14 October 2008; revised 19 May 2009
Handling editor: Alberto Apostolico

1. INTRODUCTION

The Internet is the current support for applications (net-meetings, video conferences, etc.) in which groups of users spread over a network wish to communicate intensively with privileged conditions. The users wish to obtain fast communications. The role of the provider is to interconnect the members of every group. A way to quantify the quality of a communication for a group is to consider the maximum latency between its members, i.e. the maximum distance between each of its members, called the diameter of the structure connecting the group.

1.1. General problem and related works

The network is modeled by a non-directed, weighted graph. Every group is a set of vertices of the graph. A vertex is potentially a member of several groups. Let $M_1, \ldots, M_k$ be a set of groups in a graph $G$; the problem for the provider is the following: for every group $M_i$ compute a spanning structure $T_i$ such that (1) the weight (i.e. the number of edges or the sum of the weights of edges) of the union of $T_i$ and (2) the diameter of every $T_i$ are as small as possible.

The creation of connection structures (or spanning structures) $T_i$ has already been studied in the literature. In the ‘standard’ approaches, all vertices of the graph are included in a unique group. If we consider the constraint (1) it is the well-known problem of the minimum spanning tree solved in a centralized way by Kruskal, Prim or Boruvka. If we consider the constraint (2) it is the problem of the minimum diameter spanning tree, the centralized problem is studied in [1] and a distributed algorithm is proposed in [2]. A natural generalization is to connect only a subset of vertices. If the objective is to minimize the weight of the structure, this is the well-known NP-complete Steiner tree problem [3, 4]. Several variants have been introduced. For example, minimizing criteria (1) and (2) for a unique group is studied in [5] and a distributed version is proposed in [6]. All the approaches we know concern a unique group. The originality of this article is that we consider a set of groups and simultaneously optimize the diameter of each group. Indeed, the latency of the traffic must be minimized inside each group while having a low impact on the communications of the other groups.
1.2. Our contributions

In this paper, we show that to allocate connecting structures providing an optimal maximum latency for all groups is very complex. Indeed, we prove in Section 2 that to minimize the number of links in a spanning structure or to create disjoint spanning structures for every group, each preserving the diameter, lead to hard algorithmic problems. Due to these hardness results we relax the constraint on the diameter of each group and consider a different approach in Section 3. It consists in creating a unique tree to interconnect all the groups. For this, we propose a heuristic and analyze the structures it constructs in the worst and average cases.

1.3. Notations

We introduce now the main notation used in this paper. Let \( G = (V, E, w) \) be any graph with \( V \) the set of vertices (representing the set of users), \( E \) the set of edges (representing the physical links) and \( w \) a weight positive function of the edges (\( \forall e \in E \), \( w(e) > 0 \) represents the time for a message to cross the link). We note \( d_G(u, v) \) the distance between \( u \) and \( v \) in \( G \), i.e. the sum of the weights of a path of minimum weight between \( u \) and \( v \) in \( G \). A group \( M \) is a subset of nodes (\( M \subseteq V \)). A structure \( S = (V_S, E_S) \) spanning \( M \) in \( G \) is a subgraph of \( G \) satisfying \( M \subseteq V_S \subseteq V \) and \( E_S \subseteq E \). The stretch factor induced by \( S \) on \( M \) is the ratio \( D_S(M)/D_G(M) \), where \( D_S(M) \) (resp. \( D_G(M) \)) is the diameter of group \( M \) in the structure \( S \) (resp. in the graph); \( D_S(M) = \max\{d_S(u, v) : u, v \in M\} \); this factor represents the impact on the maximum latency of communications between the members of \( M \) in \( S \) compared with the ones in \( G \).

2. TWO APPROACHES PRESERVING THE DIAMETER

In this section, we are interested by connection structures preserving the diameter (SPD) (stretch factor of 1), to obtain the best possible quality in each structure. We show that this constraint is too strong and difficult to algorithmic problems (see [7] for general references on approximation algorithms).

Definition 2.1 (SPD of a group). Let \( G = (V, E) \) be a graph and \( M \subseteq V \). The subgraph \( S = (V_S, E_S) \) of \( G \) is an SPD of \( M \) in \( G \) if \( M \subseteq V_S \subseteq V \), \( E_S \subseteq E \) and \( D_S(M) = D_G(M) \). The weight \( W(S) \) of this structure \( S \) is its number of edges: \( W(S) = |E_S| \).

2.1. Minimization of the number of edges in SPD

When several groups must be connected, it is natural to use the minimum number of links for each group. We try here, for a given group \( M \), to construct a structure preserving its diameter and using the minimum number of links. Formally, this is Problem 2.1.

Problem 2.1.

Instance: An unweighted graph \( G = (V, E) \) (or with \( w(e) = 1, \forall e \in E \)), a group \( M \subseteq V \).

Solution: An SPD \( S \) of \( M \) in \( G \) (i.e. a structure \( S = (V_S, E_S) \) spanning \( M \) such that \( D_S(M) = D_G(M) \)).

Measure: \( |E_S| \), the number of edges in \( S \).

Remark. Problem 2.1 differs from the Steiner tree problem because the diameter of the tree is constrained. As shown on the graph of Fig. 1a with group members represented by black nodes, the optimal solution to the Steiner tree problem contains three edges (see the tree given in Fig. 1b) whereas the optimal solution to Problem 2.1 contains four edges with a preserved diameter equal to two for the group (see the tree given in Fig. 1c).

Theorem 2.1. Let \( \alpha \) be any constant. An \( \alpha \)-approximation algorithm for Problem 2.1 does not exist, except if \( P = NP \).

Proof. We prove this result by contradiction showing that if a polynomial time \( \alpha \)-approximation algorithm (\( \alpha \) constant) exists for the Problem 2.1, then we can construct a polynomial time \( \beta \)-approximation algorithm (\( \beta \) constant) for the problem of minimum set cover; however this problem is not APX (i.e. not approximable by a constant); see [7]. To make the proof, we use the following standard formulation of minimum set cover.

Problem 2.2 (Minimum set cover).

Instance: \( X \) a set of elements and \( \mathcal{F} \) a set of subsets of \( X \) satisfying: \( X = \bigcup_{F \in \mathcal{F}} F \).

Solution: A subset \( C \subseteq \mathcal{F} \) covering \( X \) (\( C \) is called a cover of \( X \)): \( X = \bigcup_{F \in C} F \).

Measure: \( |C| \), the size of the cover \( C \).

A cover of minimum size is called a minimum cover. It is known that this problem is not APX; see [8].

Then the structure of our proof is the following. For any instance \((X, \mathcal{F})\) of Problem 2.2 we construct an instance \((G, M)\) of Problem 2.1 in polynomial time. Then, if there is an approximation algorithm with constant ratio for Problem 2.1, we show that we can apply it to the associated instance \((G, M)\) and extract (in polynomial time) from the returned solution (i.e. an SPD of \( M \)) a cover for \((X, \mathcal{F})\) that has a size no more than a constant times the optimal one. This means that there exists an approximation algorithm with constant ratio for Problem 2.2. This contradicts the fact that it is not APX.

Figure 1. (a) A graph with group members in filled nodes; (b) the optimal Steiner tree for the graph in (a); and (c) the optimal solution to Problem 2.1.
When $G$ and $M$ are given, we call Minimum-Weight-SPD an SPD $S$ of minimum weight.

2.1.1. Construction of the associated instance

Let $(X, \mathcal{F})$ be any instance of Problem 2.2. If there is $C_i \in \mathcal{F}$ such that $C_i = X$, then we say that $(X, \mathcal{F})$ contains a trivial solution (that is $C_i$). In the following, we only consider instances $(X, \mathcal{F})$ with non-trivial solutions. Now, for $(X, \mathcal{F})$ we construct an associated instance of Problem 2.1. This instance is such that an SPD $S$ is a Minimum-Weight-SPD of $(X, \mathcal{F})$. The group $M$ is defined as $M = X \cup \{r\}$ and we note that $m = |M|$.

The edges of $G$ are the following:

(i) Vertex $r$ is connected to each vertex of $V_1$ by a path of length $m - 1$ (using the ‘additional anonymous’ vertices).

(ii) Each vertex $u_i \in V_1$, associated to a subset $C_i$ of $\mathcal{F}$ is connected (by an edge) to all the vertices of $X$ contained in the set $C_i$.

(iii) Each vertex of $X$ is connected (by an edge) to vertex $Y$.

An example of such a transformation is given in Fig. 2. The instance of Problem 2.1 is the constructed graph $G$ and the group $M$.

For any instance $(X, \mathcal{F})$ of Problem 2.2, one can construct an associated four levels graph of Problem 2.1 by applying the previous rules. This construction is polynomial since the size of the associated graph is polynomial in the size of the instance $(X, \mathcal{F})$ of Problem 2.1.

2.1.2. Properties of SPD and Minimum-Weight-SPD in the associated instance

We now consider a four levels graph $(G, M)$. We have $D_G(M) = m$. The diameter of the group $M = X \cup \{r\}$ is between vertex $r$ and any vertex of $X$, via vertices of $V_1$. Because of the distance constraints, any SPD $S$ of $M$ in $G$ must possess at least the following edges:

(i) It must contain one edge between each vertex of $X$ and a vertex of $V_1$ (to ensure distance constraints between $r$ and vertices of $X$).

(ii) Let $a$ be the number of vertices of $V_1$ directly connected to $r$ (by a path of length $m - 1$) in $S$. Then, $S$ contains the $a(m - 1)$ edges of these $a$ (distinct) paths.

This represents a total of $(m - 1)(a + 1)$ edges. However, this number of edges is not sufficient to construct an SPD of $M$ since the distances between vertices of $X$ would not be respected (note that we consider here $(X, \mathcal{F})$ with no trivial solution). These considerations lead to the following strict lower bound. Any SPD $S$ in $G$ in an associated instance verifies

$$W(G^*_M) \leq (m - 1)(a + 1).$$

Now, we consider $G^*_M$ a Minimum-Weight-SPD of $M$ in $G$. Let $B$ be the set $(b = |B|)$ of vertices of $V_1$ that are directly connected to $r$ by paths of length $m - 1$ in $G^*_M$. As the distances between $r$ and vertices of $X$ must be $m$ in $G^*_M$, $B$ is a cover of $X$ (otherwise, there exists a vertex $u \in X$ that cannot be reached directly from $r$ and in this case, we would have $d_{G^*_M}(r, u) > m$); $G^*_M$ also contains an edge from each vertex of $X$ to at least a vertex of $B \subseteq V_1$. Note that to ensure the distance to $r$, one such edge is sufficient. However, the distance between vertices of $X$ must also be at most $m$. For that, it suffices to add the $m - 1$ edges between vertex $Y$ and vertices of $X$. From these arguments, we get

$$W(G^*_M) \leq (m - 1)(b + 2).$$

Moreover, the cover $B$ extracted from $G^*_M$ is of minimum cardinality. Indeed, otherwise, let $C^*$ be a minimum cover of $X$, with $c = |C^*| < b$. With $C^*$ one can construct an SPD $S^*$ of $M$ by connecting $r$ directly (by path of length $m - 1$) to each vertex of $C^*$, then adding any edge between any vertex of $X$ and vertices of $Y$.
a vertex of $C^*$ (it is possible since $C^*$ covers $X$) and the $m - 1$ edges between $Y$ and the $m - 1$ vertices of $X$. By counting edges, we get $W(S^*) = c(m - 1) + (m - 1) + (m - 1) = (m - 1)(c + 2) + (m - 1)(b + 1)$. By definition of $b$ and by (1) applied to the SPD $G^*_M$, we get $W(S^*) < W(G^*_M)$. This is a contradiction. The conclusion is then as follows:

The extracted cover $B$ of any Minimum-Weight-SPD

$$G^*_M$$ is a minimum cover of $X$. (3)

2.1.3. A constant approximation algorithm for Problem 2.2 from Problem 2.1.

Let $\alpha$ be a constant, $\alpha \geq 1$. Suppose that there exists an $\alpha$ approximation algorithm $Algo$ for the Minimum-Weight-SPD problem. Let $(X, F)$ be any instance of the minimum set cover problem. If $F$ contains a trivial solution, i.e. a subset $C_i = X$, then return $C_i$; otherwise, construct $G = (V, E)$ and $M \subseteq V$ the four levels graph associated to the instance $(X, F)$. Let $S$ be an SPD constructed by algorithm $Algo$ applied on $G$ and $M$, and $G^*_M$, a Minimum-Weight-SPD of $M$ in $G$. As $Algo$ is an $\alpha$-approximation algorithm, we have $W(S) \leq \alpha W(G^*_M)$. Let $A$ be the set of vertices of $V_i$ that are directly connected to $r$ in $S$ (by paths of length $m - 1$) and $a = |A|$. With (1) we have $(m - 1)(a + 1) < W(S)$. Now, let $B$ be the set of vertices of $V_i$ that are directly connected to $r$ in $G^*_M$ (by paths of length $m - 1$) and $b = |B|$. With (2) we have $W(G^*_M) \leq (m - 1)(b + 2)$. All these inequalities imply that $(m - 1)(a + 1) < \alpha(m - 1)(b + 2)$ and that $a + 1 < \alpha(b + 2)$. By (3), $B$ is a minimum cover of $X$, $B = C^*$. Hence, by applying $Algo$, one can construct a covering $A$ whose cardinality $a$ satisfies $a + 1 < \alpha(|C^*| + 2)$.

That is: $a < \alpha(|C^*| + 2\alpha - 1) \leq (3\alpha - 1)(|C^*|)$. The covering problem can then be $(3\alpha - 1)$ approximated (with $\alpha$ a constant) in polynomial time. This contradicts the fact that it is not an APX problem. \hfill \square

2.2. Edge-disjoint SPD

As the previous approach led to a problem for which there is no approximation algorithm with a constant ratio, we propose here a different approach; we study the problem to allocate SPD and using independent edges. Thus, the quality in terms of maximum communication time is guaranteed in each structure (the diameter for each group is minimum) and as the SPD are edge disjoint, the traffic of a group does not affect the traffic of other groups. However, such an edge-disjoint SPD for all groups does not always exist. We show that deciding the existence is an NP-complete problem. Formally, it is Problem 2.3.

PROBLEM 2.3.

INSTANCE: An unweighted graph $G = (V, E)$, $M_1, M_2 \subseteq V$.

QUESTION: Do two edge-disjoint SPD of $M_1$ and $M_2$ exist? i.e. does there exist $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ such that $M_i \subseteq V_i \subseteq V$, $E_1 \cap E_2 = \emptyset$, $E_i \subseteq E$ and that $D_{G_i}(M_i) = D_G(M_i)$, with $i = 1, 2$?

THEOREM 2.2. Problem 2.3 is NP-complete when $M_1 = M_2 = V$ and $D_G(M_i) = 3$, or $M_1 = M_2 = V$ and $D_G(M_i) \geq 6$, or $M_1 \cap M_2 = \emptyset$ and $D_G(M_i) = 5$, or $M_1 \cap M_2 = \emptyset$ and $D_G(M_i) \geq 8$.

Proof. To prove Theorem 2.2, we adapt and extend techniques used in [9] to prove the NP-completeness of finding disjoint rooted spanning trees of minimum depth.

Problem 2.3 is in NP. For the certificates $G_1$ and $G_2$ having a polynomial size, we can in polynomial time verify that $E_1 \cap E_2 = \emptyset$ and $D_{G_i}(M_i) = D_G(M_i)$. We prove Theorem 2.2 by reducing it to the problem of set splitting (Problem 2.4), which is NP-complete (see [8]):

PROBLEM 2.4.

INSTANCE: A hypergraph $\mathcal{H} = (V_H, C)$ ($\forall c \in C$, $c \subseteq V$).

QUESTION: Does a partition $X_1, X_2$ of $V_H$ such that $\forall c \in C$, $X_1 \cap c \neq \emptyset$ and $X_2 \cap c \neq \emptyset$ (2-colorability) exist?

Let $H = (V_H, C)$ be a hypergraph. We construct an associated graph $G$ (see Fig. 3):

Vertices of $G$: A vertex in $G$ for each vertex of $V_H$; a vertex for each hyperedge of $H$, the last set is denoted by $E$. Additional new vertices are denoted by $r$, $Y_1, Y_2, Y_{21}, Y_{21}'$. edges of $G$: Vertex $r$ is connected to all the vertices of $V_H$; vertices $Y_2, Y_{21}, Y_{21}'$ are connected to all vertices of $V_H$; $Y_1, Y_1'$ are connected to all vertices of $E$, $Y_2$ and $Y_2'$; vertex $Y_2$

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig3.png}
\caption{(a) Example of a hypergraph $\mathcal{H} = (V_H, C)$ and (b) associated graph $G = (V, E)$ to the hypergraph of (a).}
\end{figure}
is connected to \( Y_2', Y_{21}' \) and \( Y_{21} \); vertex \( Y_2' \) is connected to \( Y_{21} \) and \( Y_{21}' \); vertex \( Y_{21} \) is connected to \( Y_{21}' \); finally if \( u \in c \), then the vertex \( u \in V_H \) is connected to the vertices of \( E \) representing \( c \).

Initially, we take two identical groups \( M_1 \) and \( M_2 \), such that \( M_1 = M_2 = V \).

**Remark.** The construction of the graph is polynomial and \( D_G(M_i) = 3 \) (i.e., \( M_1 = M_2 = V \)).

We show that a solution to Problem 2.4 exists in \( G \).

2.2.1. First transformation (from Problem 2.4 to Problem 2.3)

If there is a solution \( X_1, X_2 \), \((X_i \subseteq V_H)\), to Problem 2.4 (2-colorability of \( H \)), then construct \( G_1 \) (resp. \( G_2 \)) the following SPD (see Fig. 4):

(i) Construction of \( G_1 \): For each \( u \in E \), connect it to a vertex of \( X_1 \) (there is at least one because \( X_1 \) dominates \( E \)). Connect \( r \) to all vertices of \( X_1 \); \( Y_2 \) to all vertices of \( V_H \); \( Y_1 \) to all vertices of \( V_H \); \( Y_2_1 \). Connect \( Y_1, Y_1', Y_2 \) and \( Y_{21} \); finally \( Y_21 \) to \( Y_{21}' \).

(ii) Construction of \( G_2 \): For each \( u \in E \), connect it to a vertex of \( X_2 \) (there is at least one because \( X_2 \) dominates \( E \)). Connect \( r \) to all vertices of \( X_2 \); \( Y_2' \) to all vertices of \( V_H \); \( Y_1' \) to all vertices of \( V_H \); \( Y_2' \) to \( Y_1, Y_1', Y_2 \) and \( Y_{21} \); finally \( Y_{21} \) to \( Y_{21}' \).

**Properties of \( G_i \):**

(i) \( D_{G_i}(M_i) = 3 \) (the distances between vertices of \( E \) are ensured by \( Y_1' \) and \( Y_2' \) for each structure, in the same way for \( X_1 \), with \( Y_2' \) and \( Y_2' \)). In \( G_1 \) (resp. \( G_2 \)), \( Y_21 \) (resp. \( Y_21' \)) allows \( r \) to reach \( Y_{21}' \) (resp. \( Y_{21} \)) in at most three hops, and \( \forall u \in E \), it exists a vertex of \( X_i \) which connects \( r \) to \( u \). \( G_1 \) is a SPD of \( M_i = V \).

(ii) \( E_1 \cap E_2 = \emptyset \). The two SPD \( G_1 \) and \( G_2 \) are edge disjoint.

Therefore Problem 2.3 has a solution in the associated graph \( G \).

2.2.2. Second transformation (from Problem 2.3 to Problem 2.4)

Let \( G_i = (V, E_i) \) be a solution to Problem 2.3 in the associated graph; \( D_{G_i}(M_i) = 3 \). We note that \( X_1 \), the set of vertices of \( V \) directly connected to \( r \) in \( G_i \); if some vertices of \( V_H \) are neither in \( X_1 \) nor in \( X_2 \), put them arbitrarily in \( X_1 \) or \( X_2 \). Thus \( X_1 \cap X_2 = \emptyset \) (because \( G_1 \) and \( G_2 \) are edge-disjoint structures) and \( X_i \in V_H \). Moreover \( \forall u \in E, d_{G_i}(r, u) = 2 \) (if \( r \) is not connected to a vertex \( u \in E \) by a path of length \( 2 \), via a vertex of \( V_H \), then \( d_{G_i}(r, u) \geq 3 \) and in this case \( D_{G_i}(M_i) > 3 \). Hence \( \forall u \in \mathcal{E}, \exists v \in X_i \), such that \( uv \in E_i \), and so \( X_1 \) and \( X_2 \) form a 2-coloration of \( H \).

This finishes to show that Problem 2.3 in which \( M_1 = M_2 = V \) and \( D_G(M_i) = 3 \) is NP-complete. We have now to show that the problem remains NP-complete for \( M_1 = M_2 = V \) and \( D_G(M_i) \geq 6 \). The idea of the proof is to construct an instance of graph \( G'' \) associated to the hypergraph \( \mathcal{H} \) and use the same proof with \( G'' \) instead of \( G \).

The certificate is the same as for the initial Problem 2.3. Let \( \mathcal{H} = (V_H, C) \) be a hypergraph. We construct an associated graph \( G'' \) as follows. \( G'' \) contains the graph \( G \) (constructed in the first part) plus an additional graph \( G' \) connected to vertex \( r \) of \( G \). \( G' \) is a graph whose diameter can be as large as one wants and accept two edge-disjoint structures.

The structure of graph \( G' \) depends on the desired diameter \( D_G(M_i) \). The graph \( G' \) of rank 1 is composed of two root vertices \( r' \) and \( r'' \), and a complete graph with four vertices named \( u_0, u_1, v_0 \) and \( v_1 \) (see Fig. 5b), such that the end vertices \( u_0 \) and \( v_0 \) (resp. \( u_1 \) and \( v_1 \)) are connected to the vertex \( r \) (resp. \( r' \)).

The complete graph allows us to connect all the vertices by two edge-disjoint structures. We generalize the construction of the graph \( G \) for rank \( j \) by taking \( 2(j + 1) \) vertices and two roots \( r' \) and \( r'' \) (see Fig. 5c). We connect \( u_0 \) to \( v_0 \), then \( r \) to \( u_0 \) and \( v_0 \); after this we connect each vertex \( u_k \) to \( v_k \), then \( u_k \) and \( v_k \) to \( u_{k-1} \) and \( v_{k-1} \), with \( k = 1, \ldots, j \). Finally, we connect the vertices \( u_j \) and \( v_j \).
Remark. (i) The construction of the graph \( G'' \) is polynomial.

(ii) \( D_G''(M_i) = D(G) + D(G'), \) with \( D(S) \) the diameter of the structure \( S. \)

(iii) \( D(G_i) = 3 \) and \( D(G') \geq 3 \) (in the graph \( G' \) of rank 1 we have a distance of 1 between the vertices \( u_j \) and \( v_j \), because we have a complete graph, and we must add a distance of 2 to reach the vertices \( r \) and \( r' \)).

2.2.3. First way (from Problem 2.4 to Problem 2.3)
If there is a solution \( X_1, X_2(X_i \subseteq V_H) \) to Problem 2.4, we first construct the two SPD \( G'_{1i} \) and \( G'_{2i} \) of \( G' \) (see Fig. 6):

(i) Construction of \( G'_{1i} \) of rank \( k \):

(a) take the path from \( r' \) to \( r \) via the vertices \( u_j \);
(b) connect \( v_0 \) to \( u_1 \);
(c) connect \( v_k \) to \( u_{k-1} \);

(d) connect each \( v_j \) to \( u_{j-1} \) and \( u_{j+1} \), with \( j = 1, \ldots, k-1 \).

(ii) Construction of \( G'_{2i} \) of rank \( k \):

(a) take the path from \( r' \) to \( r \) via the vertices \( u_j \);

(b) connect each \( u_j \) to \( v_j \), with \( j = 0, \ldots, k \).

Then we construct the SPD \( G''_{si} \) by adding \( G_i \) (as done in the first part of the proof while considering \( r' \) in the place of \( r \)) to \( G'_{1i} \) (see Fig. 6 for an example.)
Properties of $G''_i$:

(i) $G''_1$ and $G''_2$ are SPD of $M_1$ and $M_2$ in $G''$. Since $D_{G''}(M_i) = D(G) + D(G'_i)$ with $D(G') = D(G_i)$ because $G_i$ keeps the diameter of the group $M_i$ in $G$ (see the first part of the proof), and $D(G'_i) = D(G''_i)$ because $G'_i$ and $G''_i$ are two edge-disjoint structures in $G'$ with diameter $D(G')$ (see Fig. 6). Thus we have $D_{G''}(M_i) = D(G) + D(G''_i) = D(G_i) + D(G''_i) = D_{G''_i}(M_i)$.

(ii) $E_1 \cap E_2 = \emptyset$.

Therefore Problem 2.3 has a solution in the associated graph $G''$.

2.2.4 Second way (from Problem 2.3 to Problem 2.4)
Let $G''_i = (V, E_i)$, with $i = 1, 2$, be a solution to Problem 2.3 in the associated graph $G''$. Let $X_1$ be the set of vertices of $V_{G''}$ directly connected to $r'$ in $G''_i$. If some vertices are neither in $X_1$ nor in $X_2$, put them arbitrarily in $X_1$ or in $X_2$. Hence $X_1 \cap X_2 = \emptyset$ (because $G''_1$ and $G''_2$ are edge disjoint) and $X_1 \cup X_2 = V_{G''}$. We show that $X_1, X_2$ is a 2-coloration of $H$. We first show that

$$\forall u \in E, \quad d_{G''}(r', u) = 2. \tag{4}$$

Indeed, due to the structure of $G''$, any shortest path in $G''_i$ between $r$ and any $u \in E$ goes through $r'$. Suppose that Equation (4) is false, i.e. $\exists u \in E$ such that $d_{G''}(r', u) > 2$. In fact, $d_{G''}(r', u) > 3$ (since there is no path of length 3 between $r'$ and $u$ in $G''$).

$$\implies d_{G''}(r', u) = d_{G''}(r, r') + d_{G''}(r', u)$$

(because $d_{G''}(r, r') \geq d_{G''}(r, r')$)

$$\geq d_{G''}(r, r') + d_{G''}(r', u)$$

(by construction of $G'$)

$$= D(G) + d_{G''}(r', u)$$

(we suppose $d_{G''}(r', u) > 3$)

$$> D(G') + 3$$

$$\geq D(G'')$$

(by construction of $G''$)

$$\implies D(G''_i) > D(G'')$$

$$\implies D_{G''_i}(M_i) > D_{G''}(M_i)$$

(as $M_i = V(G'')$).

This contradicts the fact that $G''_i$ is an SPD of $G''$ for $M_i = V(G'')$. Hence Equation (4) is proved.

From Equation (4) we get $\forall u \in E, \exists v \in X_1 \subseteq V_{G''}$ such that $uv \in G''_i$. Hence any hyperedge contains an element of $X_1$ and $X_2$.

The last part of the proof concerns the case $M_1 \cap M_2 = \emptyset$, the proof remains valid also in this case. Indeed, it is sufficient to consider graph $G$ in which we connect to each vertex $u \in V$ at least a vertex of groups $M_1$ and $M_2$ (these vertices are leaves of the graph); see Fig. 7a and b. Because of these connections the diameters are increased by two units.

It is possible to generalize a part of this proof to $k$ groups, with a constant $k$. In case we have $k > 2$ groups such that

$$M_i \neq V, \quad (i = 1, \ldots, k) \text{ and } k - 2 \text{ groups that do not intersect other groups in the graph, one can use the proof of Theorem 2.2 to prove the NP-completeness of constructing $k$ edge-disjoint structures. Indeed, we can construct $k - 2$ edge-disjoint structures for the $k - 2$ groups that do not intersect other groups, and these constructions are polynomial as $k$ is a constant. For the last two groups we construct the graph $G''_i$ corresponding to a hypergraph $\mathcal{H}$, and use the proof of Theorem 2.2 to prove the NP-completeness of constructing two edge-disjoint structures. It implies that the construction of $k$ edge-disjoint structures is NP-complete in this case. Unfortunately, if we want $k$ edge-disjoint structures for $k$ groups with $M_i = V \quad (i = 1, \ldots, k)$ or $M_i \cap M_j = \emptyset$, we cannot use the proof of Theorem 2.2, and in this case it is necessary to use a reduction to the problem of hypergraph $k$-colorability.

However, as it is noticed in [10], one can get NP-hardness results for the hypergraph $k$-colorability problem, $k > 2$, for more restricted classes of hypergraphs.

3. AN APPROACH TO SOLVE THE PROBLEM

The results of Section 2 show that it is complex to find SPD. In this section we relax this constraint and propose
the following study. Given a set of groups, construct a tree spanning all groups and minimizing the diameter of each group in the tree. From the provider point of view, a tree structure has the advantage of simplifying the routing and information duplication mechanisms to propagate to the members of each group in the network. Indeed, there is a single possible path between two vertices of a tree, and the routing is easy. Moreover, as there is no cycle, the broadcast mechanism is simple, just by flooding, without complex control. In Section 3.1, we propose a heuristic to construct a tree spanning all groups and we introduce Lemmas 3.1 and 3.2, which are preliminary technical results. Then, we give in Section 3.2.1 (resp. Section 3.3) an upper bound on the maximum (resp. average) degradation of the diameter of each group induced by our heuristic. We also give in Section 3.2.2 a lower bound on the maximum degradation of the diameter for any algorithm.

3.1. Our heuristic

We present now the broad outlines of our heuristic. It takes as input the graph $G$, and the $g$ groups $M_1, \ldots, M_g$. We define a pruned tree (resp. forest) $T$ spanning any group $M$ (resp. any groups $M_1 \cup \cdots \cup M_k$) as a tree (resp. forest) in which each leaf is a member of $M$ (resp. $M_1 \cup \cdots \cup M_k$). To obtain such a pruned tree, we delete the dead branches, which are useless (a dead branch of a tree (resp. forest) $T$ is a path of vertices $u_0, \ldots, u_k$ such that for every $u_j (0 \leq j \leq k), u_j \notin M$ (resp. $u_j \notin M_1 \cup \cdots \cup M_k$) and such that $u_0$ or $u_k$ is a leaf). Our heuristic returns a pruned tree $A$ spanning the $g$ groups.

(i) Construct for each group $M_i (i = 1, \ldots, g)$ a pruned shortest paths tree, rooted in any member of the group, spanning $M_i$.

(ii) Without loss of generality let $T_1, \ldots, T_g$ be the trees sorted by increasing order of $D_G(M_i) = D_G(M_1) \leq \cdots \leq D_G(M_g)$.

(iii) Let $A_1 = T_1$. At each step $i (i = 2, \ldots, g)$, merge the trees of $A_{i-1}$ with tree $T_i$; then, break all cycles $C$ by deleting an edge $e \in C \cap T_i$; finally, delete all dead branches. The resulting forest is $A_i$.

(iv) If $A_g$ is composed of several trees, add shortest paths connecting the component vertices of $A_g$ to obtain a single pruned tree $A$ spanning $M_1 \cup \cdots \cup M_g$, otherwise $A = A_g$.

Remarks. During the construction of Forest $A_i$ we do not authorize the heuristic to modify Forest $A_{i-1}$ (this point is important in the analysis).

We now give several analytical results on the performances of our heuristic. We first prove preliminary useful lemmas.

**Lemma 3.1.** Let $m = \max\{|M_i|, i = 1, \ldots, g\}$. For all $i$, with $1 \leq i \leq g$, after $i$ iterations of the heuristic we have $D_{A_i}(M_i) \leq (m - 1) \sum_{j=1}^{i} D_G(M_j)$.

Proof. We introduce the following notation:

(i) For any tree $T = (V_T, E_T)$, we denote by $D(T)$ (resp. $W(T)$) the diameter of $T$, i.e. $D(T) = \max\{d_T(x, u) : x, y \in V_T\}$ (resp. the weight of $T$, i.e. $W(T) = \sum_{e \in E_T} w(e)$).

(ii) At step $i$ of the heuristic, we denote by $\{T_i^1, \ldots, T_i^l\}$ the set of disjoint trees of Forest $A_i$.

By definition of the heuristic, we have:

(i) For every $j, 1 \leq j \leq i$, the tree $T_j$ spanning $M_j$ is a pruned tree (see Stage (i) of the heuristic).

(ii) For every $k, 1 \leq k \leq l$, the tree $T_i^k$ is a pruned tree (see Stage (iii) of the heuristic).

Thus after $i$ iterations of the heuristic, by construction of the $T_i^j$ (see Stage (iii) of the heuristic), we have

$$\forall k, 1 \leq k \leq l, \quad D(T_i^k) \leq W(T_i^k) \leq \sum_{j=1}^{i} W(T_j).$$

Moreover, as each $T_j$ is a pruned shortest paths tree rooted in $r_j \in M_j$ spanning $M_j$, we have

$$W(T_j) \leq \sum_{u \in M_j \setminus \{r_j\}} d_G(r_j, u) \leq \sum_{u \in M_j \setminus \{r_j\}} D_G(M_j) = (|M_j| - 1) D_G(M_j).$$

Thus, we obtain

$$\forall k, 1 \leq k \leq l, \quad D(T_i^k) \leq \sum_{j=1}^{i} (|M_j| - 1) D_G(M_j). \quad (5)$$

Let $T_i^{k_0}$ be the tree of $A_i$ such that $T_i^{k_0}$ spans $M_i$ (by definition of the heuristic, a such tree does exist). Thus, $T_i^{k_0}$ is spanning $M_i$ and we have

$$D_{A_i}(M_i) = D_{T_i^{k_0}}(M_i) \leq D(T_i^{k_0}). \quad (6)$$

As $m = \max\{|M_i|, i = 1, \ldots, g\}$, by (5) and (6) we conclude

$$D_{A_i}(M_i) \leq \sum_{j=1}^{i} (|M_j| - 1) D_G(M_j) \leq (m - 1) \sum_{j=1}^{i} D_G(M_j).$$

□

To represent more precisely the disparity of the diameters in the network, we shall define and use the notion of level. As in the heuristic, the groups $M_i$ are ordered by increasing values of Diameter $D_G(M_i): D_G(M_1) \leq \cdots \leq D_G(M_g)$. Given a value $p > 1$, Level $k (k \geq 0)$ is the set of all the groups $M_i$ whose diameters $D_G(M_i)$ satisfy

$$\sum_{i \geq m} p^k D_G(M_i) \leq D_G(M_i) < p^{k+1} D_G(M_i).$$

The size of a level is the number of groups it contains (N.B. some levels may be empty). Given $p$, we note $\eta_{\max}$, the maximum size of a level.
According to Lemma 3.1 and by definition of levels, we have

We now analyze our heuristic and prove an upper bound parameter \(p\). For every \(T\) we have

\[ n_{\max} = \max\{n_i : 0 \leq k \leq K\} \]

and \(m = \max\{|M_i|, i = 1, \ldots, g\}\). Let \(l\) be the level containing the group \(M_i\).

According to Lemma 3.1 and by definition of levels, we have

\[
D_{A_i}(M_i) \leq (m-1) \sum_{j=1}^{i} D_G(M_j)
\]

\[
\leq (m-1) \left( n_1 D_G(M_1) + \sum_{k=0}^{l-1} n_k p^{k+1} D_G(M_1) \right)
\]

\[
\leq (m-1) \left( n_1 D_G(M_1) + \sum_{k=0}^{l-1} n_k p^{k+1-l} D_G(M_1) \right)
\]

\[
\left(\text{because } D_G(M_1) \leq \frac{D_G(M_1)}{p^l}\right)
\]

\[
\leq n_{\max} (m-1) D_G(M_1) \left( 1 + \sum_{k=0}^{l-1} p^{k+1-l} \right)
\]

\[
= n_{\max} (m-1) D_G(M_1) \left( 1 + \sum_{k=0}^{l-1} \frac{1}{p^k} \right)
\]

\[
\leq n_{\max} (m-1) D_G(M_1) \left( 1 + \frac{p}{p-1} \right)
\]

\[
\left(\text{because } \sum_{s=0}^{n} \left( \frac{1}{p} \right)^s \leq \frac{p}{p-1}\right).
\]

3.2. Maximum stretch factor

Given a tree \(T\) spanning \(g\) groups \(M_1, \ldots, M_g\), the degradation of the diameter of each group \(M_i\) in \(T\) is quantified by the stretch factor induced by \(T\) on \(M_i\) as follows: \(D_T(M_i)/D_G(M_i)\). Thus, a simple performance measure is the largest stretch factor value of the groups in \(T\), i.e. the maximum stretch factor induced by \(T\) as follows: \(\max\{D_T(M_i)/D_G(M_i) : i = 1, \ldots, g\}\). The aim is to construct a tree \(T\) spanning \(M_1 \cup \cdots \cup M_g\) minimizing the maximum stretch factor. We define

\[
\alpha = \min \left\{ \max \left\{ D_T(M_i) / D_G(M_i) : i = 1, \ldots, g \right\} : T \text{ tree spanning } M_1 \cup \cdots \cup M_g \right\}.
\]

3.2.1. Upper bound

We now analyze our heuristic and prove an upper bound on the maximum stretch factor of the tree returned by our heuristic. With the current notation we have the following theorem.

**Theorem 3.1.** Let \(n_{\max} = \max\{n_i : 0 \leq k \leq K\}\) and \(m = \max\{|M_i|, i = 1, \ldots, g\}\). Our heuristic constructs a tree whose maximum stretch factor is at most \(\min\{g(m-1), n_{\max} (m-1) (1 + (p/(p-1)))\}\), for every \(p > 1\).

**Proof.** (i) We first show that our heuristic constructs trees whose maximum stretch factor is always less than or equal to \(g(m-1)\).

\[
D_{A_i}(M_i) \leq (m-1) \sum_{j=1}^{i} D_G(M_j) \quad \text{(by Lemma 3.1)}
\]

\[
\leq i (m-1) D_G(M_i)
\]

(because trees \(T_1, \ldots, T_g\) are sorted by increasing order of \(D_G(M_i)\))

\[
\Rightarrow \frac{D_{A_i}(M_i)}{D_G(M_i)} \leq i (m-1)
\]

(because as \(A_i\) is a subgraph of \(A_g\) spanning \(M_i\), then \(D_{A_i}(M_i) = D_{A_g}(M_i)\))

\[
\Rightarrow \max \left\{ \frac{D_{A_i}(M_i)}{D_G(M_i)} : i = 1, \ldots, g \right\} \leq g(m-1).
\]

(ii) For every \(p > 1\) and \(\forall i, 1 \leq i \leq g\), we show the second part of Theorem 3.1.

\[
D_{A_i}(M_i) \leq n_{\max} (m-1) D_G(M_i) \left( 1 + \frac{p}{p-1} \right)
\]

(by Lemma 3.2)

\[
\Rightarrow \frac{D_{A_i}(M_i)}{D_G(M_i)} \leq n_{\max} (m-1) \left( 1 + \frac{p}{p-1} \right)
\]

\[
\Rightarrow \frac{D_{A_i}(M_i)}{D_G(M_i)} \leq n_{\max} (m-1) \left( 1 + \frac{p}{p-1} \right)
\]

(because \(A_i\) is a subgraph of \(A_g\))

\[
\Rightarrow \max \left\{ \frac{D_{A_i}(M_i)}{D_G(M_i)} : i = 1, \ldots, g \right\} \leq n_{\max} (m-1) \left( 1 + \frac{p}{p-1} \right).
\]

3.2.2. Lower bound: \(\alpha \geq g-1\)

To prove this lower bound, we consider the graph in Fig. 8. In this graph, we consider \(g\) different groups, each containing two vertices. Group \(M_1\) contains the two vertices with label 1, Group \(M_2\) contains the two vertices with label 2, \ldots and Group \(M_g\) contains the two vertices with label \(g\). All the edges with label \(\epsilon\) have weight \(\epsilon\) and all the other edges (with label 1, \ldots, \(g\)) have weight 1. Any tree \(T\) connecting the \(g\) groups is composed of all the edges of the graph except one edge with label \(i\) (with \(1 \leq i \leq g\)). Thus, the distance in \(T\) between the two members...
of group $M_i$ is $g - 1 + \varepsilon$ and we obtain $DT(M_i) = g - 1 + \varepsilon$. As $DG(M_i) = 1 + \varepsilon$, we get $DT(M_i)/DG(M_i) = (g - 1 + \varepsilon)/(1 + \varepsilon)$ which tends to $g - 1$ when $\varepsilon$ tends to 0. This proves that any algorithm applied on this graph returns a tree whose maximum stretch factor is at least $g - 1$.

We underline the fact that in the particular case where all groups have only two members (i.e. when $m = 2$), by Theorem 3.1, our heuristic constructs a tree whose maximum stretch factor is $g(m - 1) = g$. As the lower bound for any algorithm in this case (where all groups have only two members) is $g - 1$, our heuristic is almost worst case optimal for groups of size two.

Nevertheless, this ‘standard’ evaluation method (i.e. using the maximum stretch factor) is not fine enough to capture all the degradations of each group. Indeed, we can construct graphs with a maximum stretch factor of at least $g - 1$ in which only one group has an important degradation, but the other groups preserve their diameter in the graph. This is why we now evaluate our heuristic with a more precise measure: the average stretch factor.

### 3.3. Average stretch factor

We define the average stretch factor induced by a tree $T$ spanning $g$ groups $M_1, \ldots, M_g$ as $(1/g) \cdot \sum_{i=1}^{g} DT(M_i)/DG(M_i)$. With the current notation we have the following theorem.

**Theorem 3.2.** Given $G$ and $M_1, \ldots, M_g$ our heuristic constructs a tree whose average stretch factor is at most

$$\min\left\{\frac{(g + 1)(m - 1)}{2}, n_{\max}(m - 1) \left(1 + \frac{p}{p - 1}\right)\right\}, \forall p > 1.$$  

The proof of Theorem 3.2 is similar to the proof of Theorem 3.1.

**Proof of Theorem 3.2.** We first prove the upper bound $g + 1$, and then we show the second part of the theorem.

(i) According to Lemma 3.1, $\forall i$, we have

$$DA_i(M_i) \leq (m - 1) \sum_{j=1}^{i} DG(M_j),$$

$$\Rightarrow DA_i(M_i) \leq i(m - 1)$$

(because $DT_j(M_j)$ are sorted by increasing order).

$$\Rightarrow \frac{1}{g} \sum_{i=1}^{g} DA_i(M_i) \leq \frac{1}{g} \sum_{i=1}^{g} i(m - 1)$$

$$\leq \frac{m - 1}{g} \sum_{i=1}^{g} i$$

$$= \frac{(m - 1) g(g + 1)}{2}$$

$$= \frac{(g + 1)(m - 1)}{2}$$

(because $A_i$ is a subgraph of $A_g$).

(ii) For every $p > 1$ and $\forall i, 1 \leq i \leq g$, we show now the second part of the bound of Theorem 3.2. According to Lemma 3.2, we have

$$DA_i(M_i) \leq n_{\max}(m - 1) DG(M_i)$$

$$\times \left(1 + \frac{p}{p - 1}\right)$$

$$\Rightarrow \frac{DA_i(M_i)}{DG(M_i)} \leq n_{\max}(m - 1) \left(1 + \frac{p}{p - 1}\right)$$

$$\Rightarrow \frac{1}{g} \sum_{i=1}^{g} DA_i(M_i) \leq n_{\max}(m - 1) \left(1 + \frac{p}{p - 1}\right)$$

(because $A_i$ is a subgraph of $A_g$).  

**Remark.** The bound $n_{\max}(m - 1)(1 + p/(p - 1))$ is more interesting than $(m - 1)(g + 1)/2$, when $p$ is ‘large’ and $n_{\max}$ is small. This is the case when the diameters of the groups are very different from one to each other. For example, if we have groups of size 2 and if the diameter of each group is three times greater than the previous, then, by taking $p = 3$ we get $n_{\max} = 1$ and...
we obtain $n_{\text{max}}(m - 1)(1 + (p/(p - 1))) = 5/2$, whereas the bound $g + 1$ can be very large.

4. CONCLUSION AND PERSPECTIVES

In this paper we have given the results for problems related to the construction of structures spanning groups and preserving their diameter in the network.

We have given hardness results to show that these problems are hard to solve. Then, we have studied the problem of constructing a single tree spanning all the groups, in which the diameter of each group is minimized. We have proposed a heuristic to treat this problem. We have been able to give analytical results on the diameter of the groups induced in the tree.

A natural perspective is to propose an heuristic with a better upper bound, especially for the average stretch factor. However, as the links of the unique tree can be shared by several groups, we also plan to investigate (optimize and analyze) the congestion measure (i.e. the number of groups sharing the links of the tree). A third important parameter to minimize is the number of links allocated to construct the tree (i.e. its ‘cost’).

FUNDING

This work was supported by SHAMAN which is an ANR VERSO Project.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous referee and Susan Arbon for their comments.

REFERENCES