Multicast tree allocation algorithms for distributed interactive simulation

Dominique Barth[1], **Johanne Cohen**[2] **and Corentin Durbach**[1]

[1] PRiSM-CNRS, Université de Versailles-S^t Quentin en Yvelines,
45 avenue des États-Unis 78035 Versailles Cedex, France

[2] LORIA-CNRS, 615 Rue du Jardin Botanique,
54602 Villers lès Nancy, France

Abstract: We deal with a way of realizing real-time communications required by a distributed interactive simulation (DIS), using multipoint communication technics. These technics would be the basic principles of the data distributed management (DDM) of the simulation tool. We focus here on a classical interactive game between participants distributed on a geographic map, where each participant is associated to a square cell on it. The needs of communication between participants (i.e., if the associated cells overlap) are represented by a graph called *neighborhood graph*. The problem we deal with consists in covering efficiently the neighborhood graphs by groups of nodes (such that each edge is covered by at least one group), and in allocating in the target network a subtree with a given bandwidth to each group. After giving the formal definition of the considered problem, we show that it is NP-complete. Then, we give some lower bounds technics. Finally, we give two heuristics to solve this problem and we analyse them.

Keywords: Networks, distributed simulation, multicast, multipoint communication, intersection graphs, NP-completeness, heuristics, simulation.

Biographical notes: Dominique Barth is currently a full professor in the French PRiSM-CNRS laboratory in Université of Versailles, France. He received his PhD degree in computer science from Université de Bordeaux in 1994. His research interests are in multipoint communication algorithms, routing and scheduling, optical networks, approximation.

Johanne Cohen received her PhD in computer science from the University of Paris XI in 1999. She is currently at the French Centre National de la Recherche Scientifique (CRNS) and located at LORIA Laboratory (Nancy, France). Her main research interest include the study of telecommunication networks and approximation methods for hard combinatorial problems.

Corentin Durbach received his PhD in computer science from the University of Versailles in 2001. He is an engineer experienced in modeling and simulation of information system. He currently works on an European project of networks AD' HOC for the police force. He worked for operators of mobile telephony on problems of performance of the GPRS.

1 INTRODUCTION

In the last decades, the needs of the military establishment to have effective and economical tools to train personnel or to evaluate some scenarios have been a major motivation to develop virtual environments where geographically distributed participants can interact as if they are in the real combat situation (see [13, 25] and refs.). This has provided many research works in the domain of distributed interactive simulation (DIS) of discrete events, i.e., the execution of a simulation program distributed on many computers interconnected by a more or less large network (LAN or MAN). Moreover, the use of distributed simulation in commercial applications (entertainment, air traffic controller training, emergency planning exercises to deal with earthquakes) is also increasing.

A computer simulation is a computation that models the behavior of some real or imagined systems over time. Historically, the first DIS project was the SIMNET (SIMulator NETworking) project (1983-1989) [18]. The challenge now is to run an interactive application with 100.000 participants distributed on 50 sites [13]. During the simulation, the different sites have to exchange various data through the interconnection network. Whenever a simulator performs some actions that may interact with other simulators, such as moving an entity to a new location, a message is sent. Thus, the real-time constraint to overpass to make the simulation be realistic concerns the communication times.

In this paper, we mainly deal with the way of efficiently realizing simulation information exchanges. The simplest solution is to suppose that each participant, having some information to be sent to some other participants, makes a multicast of these data in all the network. Even if efficient multicast QoS-routing protocols exist [23, 8, 26], multicast is not realistic for the size of simulation considered here since it would require a too costly communication bandwidth in the network. Specifically, the distributed simulation executive must provide mechanisms to the simulators to describe both information it generates, and information it needs to receive. Two techniques can be considered in this context [13] :

- *Relevance Filtering:* data are only sent to a subset of the simulation participants. A survey on this can be found in [24].
- *Bundling:* many data may be bundled into a single packet to reduce overheads.

These communication improvement preoccupations are also the ones considered by the multipoint communication community in [1, 10, 26]. Given a group of nodes in a network, that need to share a same application, the aim is to allocate enough free bandwidth resources to them to ensure a required quality of services (in terms of communication bandwidth and delay). Usually, this consists in connecting the nodes of the group by a subtree of the network with enough bandwidth allocated on each edge [1, 5]. Then in this subtree, each node of the group sends to a same node of the subtree, called core node, data to be sent to all the other nodes; the core node broadcasts the merge of all the data it has received. Various Internet protocols support such techniques (CBT, PIM [4, 12], EXPRESS,..., [34]).

Our purpose here is to solve communication problems related to DIS (see [13, 33, 24] for motivations). The techniques that we use, are based on multipoint communication techniques, and would be a basic principle of the data distribution management of the DIS (see [28, 33]), i.e., the software in the distributed simulation controlling the distribution of information. The task of this data distribution software is to ensure that each simulator receives all the messages relevant to it, and ideally, no others.

In this context, the participants of a DIS are members of various multipoint groups such that each pair of participants having to exchange information are members of at least one common group. Then, one subtree has to be allocated in the network to each group such that bandwidth constraints are respected [5, 16]. In fact, two steps can be considered:

- At the beginning of the DIS execution, groups of participants have to be computed to allow subtrees allocation respecting bandwidth constraints.
- During the execution of the DIS, the groups and thus the corresponding subtrees allocation have

to change in real-time following the evolution of the needs of communication between participants (for example if the players of a military simulation move on the battle field).

In this paper, we focus on the first step. The second step has also been investigated in [11] by using fast local search techniques.

Objectives

We focus here on a classical interactive game between participants distributed on a geographic map called "virtual field". The participants are located in various nodes of a network (LAN and/or MAN). In these interactive games to be simulated (in a distributed way), each participant is able to make moves (character, car, plane, etc...) and these participants exchange information (position, actions, ...).

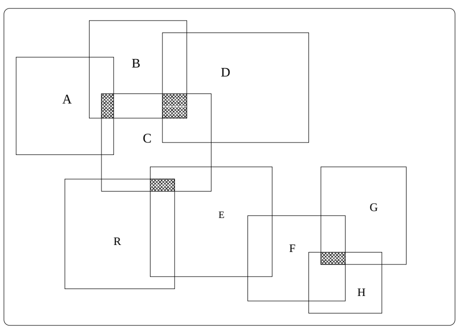


Figure 1: Virtual field

In the standard HLA (High Level Architecture) [13, 27], the virtual field is partitioned into cells [24, 27, 29, 33]. Every participant has a zone of visibility corresponding to a square zone (i.e., a box) overlapping several parts of cells (see Fig 1). It is supposed that each participant has a ray of *visibility*, marked by an arc of circle centred on the position of the participant (contained by the square zone of this participant). Inside this ray, each participant provides and receives information from the other participants. It is supposed that if two such zones overlap then both participants are close and should have to exchange data [27].

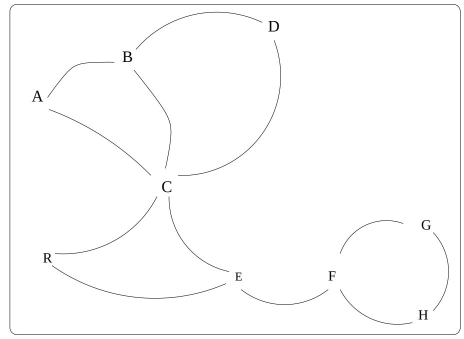
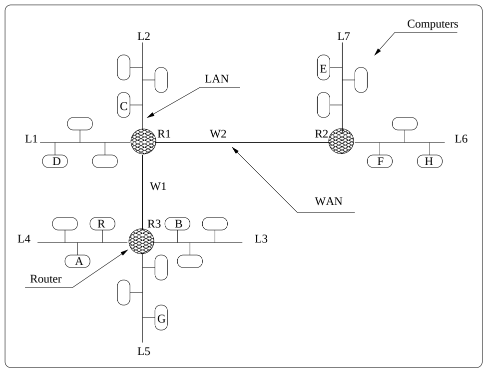


Figure 2: Neighborhood graphs H

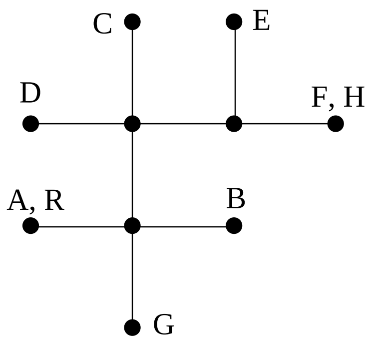
Each player (or participant) is located on a node in the target network on which runs the simulation. Fig. 3 represents the location (or mapping) of the players on the networks in our example. We assume that each player is only associated to one node of the network during the simulation (there is no migration of process towards another processor during the game).

The available bandwidth of the links of the network is limited. According to the number of participants and to the topology of the network, the generated load can saturate the links. To carry out this and to avoid the multiple broadcastings of information towards a set of participants, groups of participants are defined: each participant of a group broadcasts its information to all the other members of this group only. The global bandwidth needed by such a group is directly function of the number of members. The choice of the most suitable mode of communication is the multicast mode. Each group will be able to communicate on a multicast tree allocated to this group. In [22] a method for dynamically partitioning the participant set of a DIS into multicast groups is provided, but resource allocation to each group is not considered.

Our objective here is to obtain efficient algorithmic methods to compute groups of participants such that corresponding subtrees allocations respect bandwidth constraints of the network. The graph theory concepts are used to model the network and the virtual map.



(a)



(b)

Figure 3: Network Topology and the corresponding graph with the participant assignment

This paper is organized as follows. First, we describe the model to define the main multipoint problems that we consider. In Section 3, we prove that these problems are NP-complete and in Section 4, we give some upper and lower bound on them. Finally, we give some heuristics and experimental results.

2 Problems description

In all the following, we use graph and complexity definitions given in [7, 14].

2.1 Modelization

Neighborhood graph: as it is shown in Fig. 1 and 2, a virtual field is a particular case of d-boxes collections, as they are defined in [6]. Each element in such a collection is a d-uple

$$< (x_1, y_1),\ (x_2, y_2), \ldots, (x_d, y_d) >$$

where each pair of elements x_i and y_i are positive integers, with $x_i \leq y_i$. Such a d-uple (also called a d-box) defines an entity in a discrete d-dimensional space. We denote by $\mathcal{E}_{d,n}$ the set of all collections of d-boxes of cardinality n, where *d-boxes are labelled from* 1 *to* n. In the following, each d-box is denoted by its label.

Let $g \in \mathcal{E}_{d,n}$ be a collection. The intersection graph of g is the graph where vertices are the elements of g and where an edge links two boxes if and only if they intersect in the d-dimensional space. This kind of graphs is a particular case of intersection graphs [20, 32, 30] called *d-dimensional boxes intersection graphs* [2, 6, 30]. The set of 1-dimensional boxes intersection graphs $\mathcal{E}_{1,n}$ is the set of *interval graphs* [15]. In the case $d = 2$, this class of graphs was introduced in 1948 by Bielecki [9]. For any $d \geq 2$, the problem of knowing if a given graph is or not a d-dimensional boxes intersection graph is known to be NP-complete [19].

In this paper, a virtual field is represented by a set of 2-boxes, i.e., an element of $\bigcup_n \mathcal{E}_{2,n}$ (see Fig. 6 or Fig. 1 and 2).

Moreover, it is easy to see that many different collections in $\mathcal{E}_{d,n}$, in particular for $d = 2$, correspond to isomorphic intersection graphs. Let us call *neighborhood graphs* the intersection graphs of 2-boxes (where vertices are labelled from 1 to n) corresponding to virtual field. By definition, each edge of neighborhood graph represents an information exchange between two players.

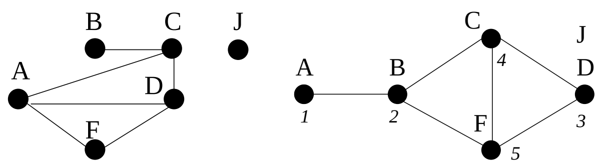
Network graph: the communication network is modeled by an undirected graph G where each edge is weighted by the bandwidth capacity of the communication link, given by a positive integer (see Fig. 3). A multicast group in such a network consists in connecting the members of the group (i.e., vertices in the graph) by a subtree.

The capacity needed by such a subtree corresponds to the resources of the network the group needs to communicate. Thus, this capacity is proportional to the number of members in the group. In our model, the capacity required by a subtree on each link it uses is equal to the cardinality of the associated multicast group. This model could be improved without changing our main approach of the problem.

2.2 Problems description

In this section, we introduce the definitions and concepts required to describe the main theoretical problem we focus on.

The set of vertices (resp. edges) of graph G is denoted by $V(G)$ (resp. $E(G)$).



(c) a neighborhood graph H

(d) a graph G and an assignment p

Figure 4: An instance of problem CEMR

Definition 1. *Let G and H be two graphs. An* assignment *from H to G is an injective mapping p from $V(H)$ onto $V(G)$.*

For example, in Fig. 4, two graphs G, and H are drawn: $V(G) = \{1, 2, 3, 4, 5\}$, $V(H) = \{A, B, C, D, F, J\}$, $E(G) = \{[1, 2], [2, 4], [2, 5], [3, 4], [3, 5], [4, 5]\}$ and, $E(H) = \{[A, C], [A, D], [A, F], [B, C], [C, D], [D, F]\}$. The assignment p from H to G is defined as follows: $p(A) = 1$, $p(B) = 2$, $p(C) = 4$, $p(D) = p(J) = 3$ and $p(F) = 5$.

For any graph G, a *collection* of G is defined as a set of subsets of vertices of G. A group is said to be a subset of vertices of G.

Definition 2. *A* covering *of graph H is a collection $\mathcal{C} = \{c_1, \ldots, c_k\}$ of H such that for every edge $[x, y] \in E(H)$, there is at least a set c_i containing x and y. We set $|\mathcal{C}| = k$ and $\bar{\mathcal{C}} = \max_{c_i \in \mathcal{C}} |c_i|$*

Considering an assignment p from H to G, for any set of vertices $V_1 \subset V(H)$, the subset $p(V_1) \subset p(V(H))$ is defined as follows : $x \in V_1$ if and only if $p(x) \in p(V_1)$. Using the same notation, if $\mathcal{C} = \{c_1, \ldots, c_k\}$ is a covering of H, then the collection $\{p(c_1), \ldots, p(c_k)\}$ of G is denoted by $p(\mathcal{C})$.

In Fig. 4, the collection $\mathcal{Z} = \{\{A, C, D\}, \{B, C\}, \{A, D, F\}\}$ is a covering of graph H (in Fig. 4(c)) because every edge of graph H is contained at most by one element in $\mathcal{Z}$. Moreovoer $p(\mathcal{Z}) = \{\{1, 2, 3\}, \{2, 4\}, \{1, 3, 5\}\}$.

Given a collection of the network graph deduced from a covering of the neighborhood graph, we define now the way communications are related to each group of the collection are realized by using allocated subtrees.

Definition 3. *Let G be a graph where the capacity of each edge α is $cap(\alpha)$ and let $\mathcal{S} = \{g_1, \ldots, g_r\}$ be a collection of G. We denote $cap(G) = \max\{cap(\alpha) : \alpha \in E(G)\}$. A* mapping *of $\mathcal{S}$ is a set $\mathcal{T} = \{T_1, \ldots, T_k\}$ of subtrees of G such that for every integer i, $1 \le i \le r$, $g_i \subset V(T_i)$.*

The load *of $\mathcal{T}$ denoted by $\epsilon_{G,\mathcal{T}}$ is defined by*

$$\epsilon_{G,\mathcal{T}} = \max_{\alpha \in E(G)} \Big(\frac{\sum_{T_i \in \mathcal{T} : \alpha \in E(T_i)} |g_i|}{cap(\alpha)} \Big).$$

The load measures the maximal ratio of resources used on a link by a mapping used to make a distributed simulation corresponding to a given neighborhood graph H on a network G.

Definition 4. *Given a collection $\mathcal{S}$ of G, if there exists a mapping $\mathcal{T}$ of $\mathcal{S}$ such that $\epsilon_{G,\mathcal{T}} \le 1$ (i.e., for every edge $\alpha \in E(G)$, $\sum_{T_i : \alpha \in E(T_i)} |g_i| \le cap(\alpha)$) then $\mathcal{S}$ is said to be* realizable *in G.*

In Fig. 4, recall that the collection $\mathcal{Z} = \{\{A, C, D\}, \{B, C\}, \{A, D, F\}\}$ is a covering of graph H. A mapping of $\mathcal{Z}$ can be $\mathcal{T} = \{T_1, T_2, T_3\}$ of subtrees of G such that $T_1 = \{[1, 2], [2, 4], [3, 4]\}$, $T_2 =$

$\{[2,4]\}$, and $T_3 = \{[1,2],[2,5],[3,5]\}$. Assume that all edges e in G are such that $cap(e) = 10$. The utilization ratio of edge $[1,2]$ is $3/10 + 3/10 (= 3/5)$ We compute the load of $\mathcal{T}$: $\epsilon_{G,\mathcal{T}} = \max(\frac{6}{10}, \frac{2+3}{10}, \frac{3}{10}) = 3/5$. So, since $1 \geq \epsilon_{G,\mathcal{T}}$, $\mathcal{Z}$ is realizable in G.

Given a network modeled by a graph G, a neighborhood graph H and an assignment of the participants of the distributed simulation on G, the main purpose here is to give a load realizable mapping on which a multipoint communication scheme could be implemented. Thus, we focus on several problems dealing with DIS communication requirements. Let us first focus on a resource allocation problem involving the network G.

Problem: EFFICIENT MAPPING (EM)
Given : Graph G, a collection S of G, a rational $k > 0$ (the ratio), and a weight $cap(a) \in N^+$, for every $a \in E(G)$.
Question : Is there a mapping $\mathcal{T}$ from S to G with a load $\epsilon_{G,\mathcal{T}} \leq k$?

Proposition 1. *Problem EM is NP-complete.*

Proof. Problem EM is obviously in NP. Moreover, consider instances (G, S, 1, cap) with S the set of all pairs of vertices of G and $cap([u,v]) = 2k$ for any $[u,v] \in E(G)$. Solving Problem EM for these instances is equivalent to decide if the edge-forwarding index of G is less or equal to k. Since answering this question is an NP-complete problem [31], Problem EM is NP-complete. □

We now define the main general problem we deal with.

Problem: COLLECTION FOR EFFICIENT MAPPING (CEM)
Given : A graph G, a weight $cap(a) \in N^+$, for every $a \in E(G)$, a neighborhood graph H, an assignment p, and a rational $k > 0$.
Question : Is there a covering $\mathcal{C}$ of H such that there is a mapping $\mathcal{T}$ from $p(\mathcal{C})$ onto G with a load $\epsilon_{G,\mathcal{T}} \leq k$?

It is easy to deduce the following corollary from Proposition 1.

Corollary 1. *Problem CEM is NP-complete.*

In Problem CEM, each group has a possibility to use every spanning tree for communications. For example, in Fig. 4, group $\{A,F,D\}$ can be mapped by tree $\{[1,2],[2,5],[5,3]\}$ or by tree $\{[1,2],[2,4],[4,5],[5,3]\}$. In fact, in many protocols, we have to deal with the routing function of the network to exchange packets. A routing function r for the network G is a mapping from $V(G)^+$ to $E(G)^+$, i.e, for every subset S of vertices, $r(S)$ is a tree which covers all the vertices of S. We thus focus on a particular case of problem CEM.

Problem: COLLECTION FOR EFFECTIVE MAPPING WITH ROUTING (CEMR)
Given : A graph $G = (V,E)$, a weight function $cap : E \to N$, a routing function r of G, a graph H of neighborhood, an assignment function p, a rational $k > 0$.
Question : Is there a covering $\mathcal{C} = \{c_1,\ldots,c_t\}$ of H such that $\mathcal{T} = \{r(c_1),\ldots,r(c_t)\}$ is a mapping from $p(\mathcal{C})$ to G with load $\epsilon \leq k$?

This problem is NP-complete and the next section is devoted to prove this fact.

3 Complexity of Problem CEMR

Theorem 1. *Problem CEMR is NP-complete, even if graph G is a tree.*

Proof. First, problem CEMR is in NP because checking whether a given covering $\mathcal{C}$, $\mathcal{T} = \{r(c_1),\ldots,r(c_t)\}$ is a mapping from $p(\mathcal{C})$ to G with load less than k, can be performed in polynomial time. Now, we transform a known NP-complete problem called 3-PARTITION to CEMR. Problem 3-PARTITION is defined as follow [14]:·

Problem: PROBLEM 3-PARTITION
Given : A set $S = \{s_1,\ldots,s_{3m}\}$, a function

$w : S \to N$, an integer m.
Question : Can S be partitioned into m disjoint sets $\{S_1, \dots, S_m\}$ such that for $1 \le i \le m$, $\sum_{c \in S_i} w(c) = 1/m * \sum_{c \in S} w(c)$?

Consider an instance $\mathcal{I}$ of 3-PARTITION : a set $S = \{s_1, \dots, s_{3m}\}$, a function $w : S \to N$, and an integer m. We shall construct an instance of problem CEMR such that the desired mapping exists if and only if S can be partitioned into m disjoint sets $\{S_1, \dots, S_m\}$ such that for $1 \le i \le m$, $\sum_{c \in S_i} w(c) = 1/m * \sum_{c \in S} w(c)$.

Let $\beta = \frac{\sum_{c \in S} w(c)}{m}$ and $q = 3m$. G is a tree such that it contains

- a vertex u;
- m paths $X_1, \dots, X_m$ of $q + 1$ vertices such that, for i, $1 \le i \le m$, $X_i = \{u, X_i(1), \dots, X_i(q)\}$. The capacity of all the edges of X_i, $1 \le i \le m$ is $2q + 3m^2\beta$. Each of theses paths corresponds to one partition set;
- q paths $Y_1, \dots, Y_q$ such that for $1 \le j \le q$, $Y_j = \{u, z_j(1), \dots, z_j(m), Y_j(1), \dots, Y_j(3m^2 w(s_j) - m)\}$. The capacity of all the edges in $\{[Y_j(\ell), Y_j(\ell + 1)], [z_j(m), Y_j(1)] : 1 \le \ell \le 3m^2 w(s_j) - m - 1\}$ is equal to $3m^2 w(s_j)) + 1$. Moreover, the capacity of all the edges in $\{[z_j(\ell), z_j(\ell + 1)], [u, z_j(1)] : 1 \le \ell \le m - 1\}$ is equal to $3m^2 w(s_j) + 2m - 1$. Each of theses paths corresponds to one element of set S.

Note that paths $X_1, \dots, X_m$ (resp. $Y_1, \dots, Y_q$) represent one element of the partition (resp. S).

As graph G is a tree, it is easy to determine the routing function r. If A is a subset of nodes of G, then $r(A)$ corresponds to smallest subtree of G such that all its leaves are nodes of A.

Now, we will build the neighborhood graph H and the function of associated assignment p in the following way: each element s_j of collection S is represented by a connected graphs called H_j such that

- $\{a_j^i, b_j^i, h_j(\ell) : 1 \le i \le m \wedge 1 \le \ell \le 3m^2 w(s_j) - m\}$ is $V(H_j)$. Moreover, for any $1 \le i \le m$, for any $1 \le \ell \le 3m^2 w(s_j) - m$, $p(a_j^i) = z_j(i)$, $p(b_j^i) = X_i(j)$ and $p(h_j(\ell)) = Y_j(\ell)$.
- $E(H_j) = E_1 \cup E_2 \cup E_3$ such that
  $E_1 = \{[a_j^i, h_j(\ell)] : 1 \le i \le m \wedge 1 \le \ell \le 3m^2 w(s_j) - m\}$
  $E_2 = \{[h_j(\ell'), h_j(\ell)] : 1 \le \ell, \ell' \le 3m^2 w(s_j) - m\}$,
  $E_3 = \Big\{[a_j^{\ell'}, a_j^{\ell}], [a_j^{\ell}, b_j^{\ell}] : 1 \le \ell, \ell' \le m\Big\}$.

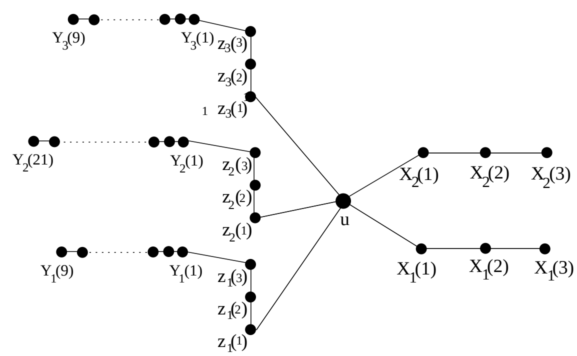


Figure 5: Tree G building from an instance of problem 3-PARTITION such that $S = \{s_1, s_2, s_3\}$, $m = 2$ and $w(s_1) = 1$, $w(s_2) = 2$, $w(s_3) = 1$.

By construction, since for any $1 \le j \le q$, H_j is composed of a complete graph and a set of edges, it is easy to see that H_j is an intersection graph. Thus, since graph H is a union of q connected graphs $H_1, \dots, H_q$, graph H is a neighborhood graph.

The construction of our instance of Problem CEMR is completed by setting $k = 1$. As the problem 3-PARTITION is NP-complete in strong sense, it is easy to see that the number of vertices of G and H is $O(m^3\beta)$. Our construction can be accomplished in polynomial time.

Now, the remainder of the proof is devoted to show the following property :

Property 1. *There is a partition $\mathcal{S} = \{S_1, \dots, S_m\}$ of S such that, $1 \le i \le m$, $\sum_{c \in S_i} w(c) = \beta$ if and only if there exists a covering $\mathcal{C} = \{c_1, \dots, c_t\}$ of H such that $\mathcal{T} = \{r(c_1), \dots, r(c_t)\}$ is a mapping of $p(\mathcal{C})$ in G with a load $\epsilon \le k$.*

First, we consider a partition $\mathcal{S} = \{S_1, \dots, S_m\}$ of S such that $1 \le i \le m$, $\sum_{c \in S_i} w(c) = \beta$. We will construct a covering $\mathcal{C} = \{c_1, \dots, c_t\}$ of H such that $\mathcal{T} = \{r(c_1), \dots, r(c_t)\}$ is a mapping from $p(\mathcal{C})$

to G with a load $\epsilon \leq k$. Every element s_j where $1 \leq j \leq q$ is represented by m elements of $\mathcal{T}$, $c_{i,j}$, with $1 \leq i \leq m$.

- If element s_j is in S_i, then $c_{i,j}$ is equal to $\{b^i_j, a^t_j, h_j(\ell) : 1 \leq \ell \leq 3m^2w(s_j) - m \wedge 1 \leq t \leq m\}$. Note that $|p(c_{i,j})| = 3m^2w(s_j) + 1$.
- If element s_j is not in S_i, then element $c_{i,j}$ contains vertices a^i_j and b^i_j : $|p(c_{i,j})| = 2$.

Let us compute the load of this mapping $\mathcal{T}$.

- Let j be an integer between 1 and q. We focus on edge $e = [u, z_j(1)]$ of path Y_j. Only elements c in $A = \{c_{\ell,j} : 1 \leq \ell \leq m\}$ are such that $e \in r(c)$. Thus, $\sum_{c\in A} |p(c)| = 3m^2w(s_j) + 2m - 1$. Let consider the following ratio :

$$\frac{\sum_{c\in A} |p(c)|}{cap(e)} = \frac{3m^2w(s_j) + 2m - 1}{3m^2w(s_j) + 2m - 1} = 1$$

  We can apply the same argument with all the edges of path Y_j.
- Let i be an integer between 1 and m. We focus on all the edges of path X_i. Without loss of generality, we consider edge $e = [u, X_i(1)]$. Only elements $c_{i,\ell}$, $1 \leq \ell \leq q$ are such that $e \in r(c_{i,\ell})$. If $s_\ell \notin S_i$, then we have $|p(c_{i,\ell})| = 2$. Otherwise (i.e if $s_i \in S_\ell$), $|p(c_{\ell,i})| = 3m^2w(s_\ell) + 1$. Let consider the following ratio :

$$\begin{aligned}\frac{\sum_{c\in\left\{c_{i,\ell},1\leq\ell\leq q\right\}} |p(c)|}{cap(e)} &= \frac{2(q-|S_i|)+3m^2\beta+|S_i|}{2q+3m^2\beta} \\ &\leq 1\end{aligned}$$

Since $\epsilon_{G,\mathcal{T}} = \max_{\alpha\in E(G)} \left(\frac{\sum_{T_i\in\mathcal{T}:\alpha\in E(T_i)} |g_i|}{cap(\alpha)}\right)$, the load of this mapping is less or equal to 1. We have proved the first part of Property 1.

Conversely, let us consider a covering $\mathcal{C} = \{c_1, \ldots, c_t\}$ of H such that set $\mathcal{T} = \{r(c_1), \ldots, r(c_t)\}$ is a mapping from $p(\mathcal{C})$ to G with a load $\epsilon_{G,\mathcal{T}} \leq 1$.

Let j be an integer such that $1 \leq j \leq q$. Edge $e = [h_j(3m^2w(s_j) - m), h_j(3m^2w(s_j) - m - 1)]$ and vertex $v = h_j(3m^2w(s_j) - m)$ are considered. In graph H, vertex v is adjacent to vertices in set A such that $A = \{a^i_j, h_j(\ell) : 1 \leq \ell \leq 3m^2w(s_j) - m - 1 \wedge 1 \leq i \leq m\}$. Since $|p(A)| = 3m^2w(s_j) - 1$, and since the capacity of e equals to $3m^2w(s_j) + 1$, there is at least two elements c^1, c^2 in $\mathcal{T}$ such that $e \in r(p(c^1))$, $e \in r(p(c^2))$, $v \in c^1 \cap c^2$, $A \cup \{v\} \subseteq c^1 \cup c^2$ and $c^1 \neq c^2$. By a listing of cases, we can check that $c^1 = c^2$ and that there is only one element c_j in $\mathcal{T}$ such that $e \in r(p(c_j))$.

Let $B = \{[u, z_j(1)], [z_j(\ell), z_j(\ell + 1)] : 1 \leq \ell \leq m - 1\}$ be a set of edges. The capacity of all theses edges is $3m^2w(s_j) + 2m - 1$. Without loss of generality, we focus on edge $e' = [z_j(1), z_j(2)]$. Let $D = \{c : c \in \mathcal{T} \wedge e' \in r(c)\}$. Note that since $e' \in r(p(c_j))$, $c_j \in D$. Moreover, since there are edges between vertices b^i_j and $a^{i'}_j$ in H and $e' \in r(p(\{b^i_j, a^{i'}_j\}))$ with $1 \leq i, i' \leq m$, we have $\{b^i_j : 1 \leq i \leq m\} \in \bigcup_{c\in D} c$. Now, we will prove by contradiction that there exists at most one element b in $\{b^i_j : 1 \leq i \leq m\}$ such that $b \in c_j$. So, we assume that c_j does not contain any vertex in $\{b^i_j : 1 \leq i \leq m\}$. So there exist some elements $c'_1, \ldots, c'_{x_j}$ in $\mathcal{T} \cap D$ such that they cover edges $\{b^i_j, a^i_j : 1 \leq i \leq m\}$ in H. By definition, we have $\sum_{\ell=1}^{x_j} |p(c'_\ell)| \geq 2m$. Since the capacity of e' equals to $3m^2w(s_j) + 2m - 1$ and since $|p(c_j)| \geq 3m^2w(s_j)$, we have

$$1 \geq \epsilon_{G,\mathcal{T}} \geq \frac{\sum_{c\in D} |p(c)|}{cap(e')}$$

$$3m^2w(s_j) + 2m - 1 \geq \sum_{c\in D} |p(c)| \geq 3m^2w(s_j) + 2m.$$

The previous inequality leads to a contradiction. So at most one element b in $\{b^i_j : 1 \leq i \leq m\}$ is such that $b \in c_j$.

Now, we will construct a collection $\mathcal{S} = \{S_1, \ldots, S_m\}$ of S as follows : $s_j \in S_i$ if all elements in set $\{h_j(\ell), a^{\ell'}_j, b^i_j : 1 \leq \ell' \leq m \wedge 1 \leq \ell \leq 3m^2w(s_i) - m\}$ belong to the same element $c_{i,j}$ in covering $\mathcal{C}$. Let $A_\ell = \{c : [u, X_\ell(1)] \in r(c)\}$ for $1 \leq \ell \leq m$. First, we will prove by contradiction that $\mathcal{S}$ is a partition of S using the capacities of edges $[u, X_\ell(1)]$, for $1 \leq \ell \leq m$. So, we assume that there exists $s_j \in S$ such that $s_j \in S_i$ and $s_j \in S_{i'}$ with $i \neq i'$ and $1 \leq i, i' \leq m$. By construction of $\mathcal{S}$, we have

$$3m^2\beta + 2q \geq \sum_{c \in A_\ell} |p(c)| \geq \sum_{s_j \in S_\ell} 3m^2 w(s_j) \qquad (1)$$

We focus on a value of $\sum_{\ell=1}^{m} \sum_{s \in S_\ell} w(s)$. From Inequality 1, we have

$$\sum_{\ell=1}^{m} \sum_{s_i \in S_\ell} 3m^2 w(s_i) \leq \sum_{\ell=1}^{m} \sum_{c \in A_\ell} |p(c)| \leq 3m^2(m\beta) + 2qm \qquad (2)$$

Since $s_j \in S$ such that $s_j \in S_i$ and $s_j \in S_{i'}$ with $i \neq i'$ and $1 \leq i, i' \leq m$.

$$\begin{aligned} \sum_{s \in S} 3m^2 w(s) + 3m^2 w(s_j) &\leq \sum_{\ell=1}^{m} \sum_{s_i \in S_\ell} 3m^2 w(s_i) \\ &\leq 3m^2(m\beta) + 2qm \end{aligned} \qquad (3)$$

Since $3m^2 w(s_j) > 2mq$, Inequality 3 leads to a contradiction. So $\mathcal{S} = \{S_1, \ldots, S_m\}$ is a partition of S. Moreover, from Inequalities 1 and 3, it is easy to deduce that $\sum_{s_j \in S_i} w(s_j)$ equals to β. Thus, $\mathcal{S}$ is a partition satisfying the following property : for every element S_ℓ of $\mathcal{S}$, $\sum_{s_j \in S_i} w(s_j) = \beta$.

We have proved Property 1 and the proof of Theorem 1 is completed.

□

The proof of Theorem 1 proves that Problem CEMR remains NP-complete even if graph G is a tree. In this case, Problem CEMR is equivalent to Problem CEM. Moreover, the proof of Theorem 1 works because graph G contains several paths. But it does not work if graph G is a path. However, we believe that the following conjecture is valid.

Conjecture 1. *Problem CEMR is solvable in polynomial time for the case where G is a path.*

4 Lower bounds of Problem CEMR

This section is devoted to giving lower bounds for problem CEMR. We deal here with the optimization problem corresponding to Problem CEMR, i.e., minimizing the load.

Definition 5. *Let H be a digraph and $\mathcal{C}$ be a covering of H. The covering $\mathcal{C} = \{C_1, \ldots, C_k\}$ is* minimal *if and only if there is no $C_i \in \mathcal{C}$ such that there exists $c \in C_1$ and $\{C_1, \ldots, C_i - \{c\}, \ldots, C_k\}$ is a covering of H.*

Covering $\{\{A,C,D\},\{A,B,C\},\{A,D,F\}\}$ in Fig. 4, is not minimal because $\{\{A,C,D\},\{B,C\},\{A,D,F\}\}$ is still a covering (and it is minimal). It is clear that to study Problem CEMR, we only have to consider minimal coverings. Let us denote by $\mathcal{C}_m(H)$ the set of all minimal coverings of H and by $\mathcal{P}(S)$ the set of all the partitions of a given set S.

Lemma 1. *For any digraph H we have $|\mathcal{C}_m(H)| \leq |\mathcal{P}(E(H))|$ and for any $\mathcal{C} \in \mathcal{C}_m(H)$ we have $|\mathcal{C}| \leq |E(H)|$.*

Proof. Let us consider any total order on the vertices of H. Each minimal covering of H can be considered as a lexicographically ordered set based on the total order. We define the following injective function $\mathcal{I}$ from $\mathcal{C}_m(H)$ into $\mathcal{P}(E(H))$. Consider the ordered set $\mathcal{C} = \{c_1, \ldots, c_k\} \in \mathcal{C}_m(H)$: $\mathcal{I}(\mathcal{C})$ is obtained by Algorithm in Table 1 (In Fig. 4, $C = \{\{A,C,D\},\{A,D,F\},\{B,C\}\}$ is a minimal covering and $\mathcal{I}(C)$ is equal to $\{\{[A,C],[A,D],[C,D]\},\{[A,F],[D,F]\},\{[B,C]\}\}$).

Since $\mathcal{C}$ is a minimal covering, then for each $P_i \in \mathcal{I}(\mathcal{C})$, $P_i \neq \emptyset$. Thus, by definition of $\mathcal{I}$, it is clear that $\mathcal{I}$ is an injective function and that $\mathcal{I}(\mathcal{C})$ is a partition of $E(H)$ with cardinal $|\mathcal{C}|$. □

Considering an instance (G, cap, r, H, p, k) of Problem CEMR, if a covering $\mathcal{C}$ of H is such that $p(\mathcal{C})$ is realizable (see Definition 4), then for any $c \in \mathcal{C}$, $|c| \leq cap(G)$. To enumerate all the coverings having this property, we could think of generate all the partitions of $E(H)$ where the greatest size of a part is less than $cap(G)^2$ (i.e., the number of arcs in a clique with $cap(G)$ vertices). Unfortunately, even if $cap(G)$ is bounded by a constant c, the number of partitions to be generated is in $\theta(e^{p(|E(H)|)})$, with $p(x)$ a polynomial function. So generating all the partitions can not be performed in polynomial time.

Table 1: An algorithm to compute an injective function $\mathcal{I}$ in Lemma 1

INPUT: *An ordered set* $\mathcal{C} = \{c_1, \dots, c_k\} \in \mathcal{C}_m(H)$
OUTPUT: *A value* $\mathcal{I}(\mathcal{C})$.
ALGORITHM:

1. $\mathcal{I}(\mathcal{C}) \leftarrow \{P_1, \dots, P_k\}$ with $P_i = \emptyset$
2. For $i = 1$ to k do

    For each edge $[u, v]$ of H not considered yet, such that $u, v \in c_i$, do

        $P_i \leftarrow P_i \cup \{[u, v]\}$

3. return $\mathcal{I}(\mathcal{C})$

In most cases, $|\mathcal{C}_m(H)| \ll |\mathcal{P}(E(H))|$. Even if $cap(G) = c$, we don't know if $|\mathcal{C}_m(H)|$ is or not a polynomial function of $|E(H)|$.

Problem CEMR is also related to different problems of mapping and cuts in graphs. This leads us to give bounds about the maximal size of a group in a covering of H that can be efficiently mapped in G.

Let (G, cap, H, p, k) be an instance of Problem CEM and consider $V_1 \subset V(G)$ a vertex subset of G (note that we deal here with Problem CEM because the lower bounds can be given. But this problem does not on the routing function). We define

- the G-cut of V_1 by $c_G(V_1) = \{[u, v] \in E(G) : u \in V_1, v \in V(G) - V_1\}$,
- the H-cut of V_1 by $c_H(V_1) = \{[x, y] \in E(H) : p(x) \in V_1, p(y) \in V(G) - V_1\}$.

For example, in Fig. 4, let V_1 be $\{4, 5\}$. $c_G(V_1)$ (resp. $c_H(V_1)$) is equal to $\{[2, 4], [2, 5], [3, 5], [4, 3]\}$ (resp. $\{[A, C], [A, F], [B, C], [C, D], [D, F]\}$).

Proposition 2. *Let (G, cap, H, p, k) be an instance of Problem CEM. Consider $\mathcal{C}$ a covering of H such that $p(\mathcal{C})$ is realizable in G. Then, the size of each element c of $\mathcal{C}$ should satisfy the following property:*

$$2 \cdot \min\left(\frac{d_{G,H}}{\Delta_H}, \sqrt{d_{G,H}}\right) \leq \max_{c \in \mathcal{C}} |c| \leq cap(G) \ ,$$

with $d_{G,H} = \max_{V_1 \subset V(G)} \frac{|c_H(V_1)|}{|c_G(V_1)|}$.

Proof. The upper bound is straight forward, We only deal with the lower bound. Let (G, cap, H, p, k) be an instance of Problem CEM. Consider a vertex subset V_1 of G and let α be an edge in $c_G(V_1)$. This edge can be used by the subtree allocated to a group containing at most $cap(\alpha)$ vertices. The maximal number of links of $c_H(V_1)$ used by such a group is the maximal number of edges in a bipartite graph with $cap(\alpha)$ vertices and with maximal degree Δ_H, i.e., with at most $\frac{cap(\alpha)}{2} \cdot \min(\Delta_H, \frac{cap(a)}{2})$ edges.

Thus, the maximal number of edges in $c_H(V_1)$ that then can be used by groups using edges in $c_G(V_1)$ is $\sum_{\alpha \in c_G(V_1)} \frac{cap(\alpha)}{2} \cdot \min(\Delta_H, \frac{cap(a)}{2})$. So we can state that:

Claim 1. *If there exists a covering $\mathcal{C}$ of H such that $p(\mathcal{C})$ is a grouping that can be mapped on G, then $\forall\ V_1 \subset V(G)$,*

$$|c_H(V_1)| \leq \sum_{\alpha \in c_G(V_1)} \frac{cap(\alpha)}{2} \min(\Delta_H, \frac{cap(a)}{2}).$$

Let $\mathcal{C}$ be a covering of H such that $p(\mathcal{C})$ is realizable and consider V_1 a vertex subset of G. The maximal size of a group using an edge α in $c_G(V_1)$ is thus equal to $\min(cap(\alpha), \max_{c \in \mathcal{C}} |c|)$. From Claim 1, we conclude that $\forall\ V_1 \subset V(G)$,

$$|c_H(V_1)| \leq |c_G(V_1)| \cdot \frac{M_\mathcal{C}}{2} \cdot \min(\Delta_H, \frac{M_\mathcal{C}}{2}) \ ,$$

with $M_\mathcal{C} = \max_{c \in \mathcal{C}} |c|$. The lower bound given in the proposition is a direct consequence of this inequality. □

From Proposition 2, parameter $d_{G,H}$ helps us to give a good idea on a lower bound of the cardinality of a cover. However,

Lemma 2. *Consider an instance (G, cap, H, p, k) of Problem CEM. Computing $d_{G,H} = \max_{V_1 \subset V(G)} \frac{|c_H(V_1)|}{|c_G(V_1)|}$ is a NP-complete problem.*

Proof. Let us consider the case of $|V(G)| = |V(H)|$, p being a bijective function and H a complete graph (which is a possible neighborhood graph).

In this case, $d_{G,H} = \max_{V_1 \subset V(G)} \frac{|V_1|(|V(G)|-|V_1|)}{|c_G(V_1)|}$. From [21] we know that, given a graph G, computing $\max_{V_1 \subset V(G)} \frac{|V_1|(|V(G)|-|V_1|)}{|c_G(V_1)|}$ is a NP-complete problem. □

Table 2: Algorithm to compute a lower bound of parameter $d_{G,H}$

INPUT: *A graph G and a graph H*
OUTPUT: *a real LowerB*
ALGORITHM:

1. $i \leftarrow 1$; $G' \leftarrow G$; $H' \leftarrow H$
2. For $i = 1$ to k do
3. While $V(H') \neq \emptyset$ do

   Let $u \in V(G')$ such that $w_{H',G'}(u)$ is maximum

   - $\mathcal{N}(u) \leftarrow i$
   - $i \leftarrow i + 1$
   - $H' \leftarrow H' - p^{-1}(u)$; $G' \leftarrow G' - \{u\}$
4. While $V(G') \neq \emptyset$ do

   Let $u \in V(G')$ such that $\Delta_{G'}(u)$ is maximum

   (a) $\mathcal{N}(u) \leftarrow i$

   (b) $i \leftarrow i + 1$

   (c) $G' \leftarrow G' - \{u\}$
5. $LowerB \leftarrow 0$
6. For each $i \in [1, \ldots, |V(G)|]$ do

   (a) $V_i' \leftarrow \{u \in V(G) : \mathcal{N}(u) \leq i\}$

   (b) $LowerB \leftarrow \max(LowerB, \frac{|c_H(V_i')|}{|c_G(V_i')|})$
7. return $LowerB$.

In order to compute good lower bounds for $d_{G,H}$, we now give an heuristic (see Table 2 for a formal description).

We first give a *labelling $\mathcal{N}$ of the vertices of G* verifying the following property: let H' be a subgraph of H and G' be a subgraph of G. Consider u a vertex in G'. We define $w_{H',G'}(u) = \frac{|\{[x,y] \in E(H') :\ x \in p^{-1}(u),\ \ y \notin p^{-1}(u)\}|}{\Delta_{G'}(u)}$. The lower bound of $d_{G,H}$ we compute here is $\max_{1 \leq i \leq |V(G)|} \frac{|c_H(V_i')|}{|c_G(V_i')|}$, with $V_i' = \{u \in V(G) :\ \mathcal{N}(u) \leq i\}$ (see Algorithm described in Table 2).

Notice that the lower bound of Proposition 2 is relevant only if H is dense in comparison with G. The following result is directly deduced from Proposition 2 and from Lemma 1.

Corollary 2. *For any instance (G, cap, H, r, p, k) of Problem CEMR, the set of possible solutions can be generated by the enumeration of all the partitions of $E(H)$ with maximal size of a part between $2 \cdot \min\left(\frac{d_{G,H}}{\Delta_H}, \sqrt{d_{G,H}}\right)$ and $cap(G)$, with $d_{G,H} = \max_{V_1 \subset V(G)} \frac{|c_H(V_1)|}{|c_G(V_1)|}$.*

Notice that Corollary 2 allows us to improve our algorithm which enumerates all the possibles solutions in order to see the quality of our heuristics given in the next section.

5 Heuristics

This section is devoted to describing two heuristics we propose to solve Problem CEMR. Given an instance of Problem CEMR, these heuristics return a covering of H.

The first heuristic uses a natural idea. Considering first a covering in which each element is an edge of H, the heuristic (randomly) merges elements to (locally) decrease the current load.

The main interest of the second heuristic is to give a first (not optimized) application of a strong property of neighborhood graphs. The second heuristic computes a covering consisting in maximal cliques of the neighborhood graphs and then optimizes the intersection of these elements of the covering.

5.1 An edge-covering heuristic for Problem CEMR.

We give here a simple heuristic called Heuristic 1 performing as follows (see Table 3 for a formal description). At the beginning, all the edges of H are the elements of covering of H (see Definition 2). The load of the current solution is computed (see Definition 3). At each step of this heuristic, we modify the current solution by merging two elements of the current covering as follows: a group g is randomly chosen and the set of the groups of the current solution such that all the groups have at least one common vertex with g is computed. For each group g' of this set, we compute the load of a solution in which groups g and g' are merged. And among all new solutions, the current solution is the solution with minimum load. This process is reiterated until there is no way to modify the current solution.

For example, we consider an instance of problem CEMR with a neighborhood graph H of 6 vertices (see Fig. 4(c)), a graph G of the target instance of CEMR and an assignment p where $p(A) = 1$, $p(B) = 2$, $p(C) = 4$, $p(D) = p(J) = 3$ and $p(F) = 5$ (see Fig. 4(d)). We apply Heuristic 1 by considering that all edges in G have the same capacity equal to 10. After the initialization phase, each edge of H is a group (one communication per group). The current solution is $\{\{A, C\}, \{A, D\}, \{A, F\}, \{B, C\}, \{C, D\}, \{D, F\}\}$. Now, we will try to merge some groups without increasing the load of the current solution. Assume that the routing function r returns a steiner tree given a set of vertices and $r(\{A, D\})) = \{[1, 2], [2, 5], [5, 3]\}$. The load of edge $[3, 4]$ is 2, the load of edges $[2, 4]$, $[2, 5]$, $[3, 5]$ is 4 and the load of edge $[1, 2]$ is 6. A group $\{A, D\}$ is randomly chosen and the groups $\{A, F\}, \{A, C\}, \{C, D\}, \{D, F\}$ have one common vertex with group $\{A, D\}$. Now, we compute the load of a new covering by merging $\{C, D\}$ with group $\{A, D\}$: so we compute the load of solution $\{\{A, C, D\}, \{A, F\}, \{B, C\}, \{D, F\}\}$ (note that group $\{A, C\}$ is removed because it is a subset of $\{A, C, D\}$). Its load is equal to 5/10. We do the same process for groups $\{A, F\}, \{A, C\}$, $\{D, F\}$. We keep the best solution which consists of merging $\{C, D\}$ with group $\{A, D\}$. Again, a group $\{B, C\}$ is randomly chosen. Only group $\{A, C, D\}$ has one common vertex with group $\{B, C\}$ and the solution obtained by merging group $\{B, C\}$ with $\{A, C, D\}$ has 6/10 as load. So group $\{B, C\}$ can not merged without increasing load and the current solution is not modified at this stage. And so on. At the end, Heuristic 1 gives the solution $\{\{A, C, D\}, \{A, F\}, \{B, C\}, \{F, D\}\}$ (it is easy to see that merging any two groups does not improve the quality of the solution).

5.2 A clique-covering heuristic for Problem CEMR

We will describe an heuristic based on some properties of a neighborhood graph. In Section , we notice that neighborhood graphs are isomorphic to intersection graphs. Neighborhood graphs verify the Helly property [30]. This property given by Helly in 1923, says that, given n convex subsets in a d-dimensional Euclidean space with $n \geq d + 1$, if each collection of $d + 1$ subsets contains a same common point, then there is a same point common to the n subsets. Thus in case of neighborhood graphs (i.e., $d = 2$), if a set of at least 3 vertices induces a clique then the corresponding boxes in the virtual field have a no empty intersection.

We know that if an intersection graph with n vertices verifies this property then this graph contains a polynomial number of induced cliques, and that these cliques can be enumerated in time $O(n^7)$ [3, 30]. Thus, the CLIQUE-MAX problem can be solved in polynomial time for intersection graphs of d-dimensional boxes, with d bounded by a given constant c.

We now give an improved result to obtain the heuristic for our main problem. We will give an algorithm to enumerate all the maximal induced cliques of a neighborhood graph in time $O(n^4)$ where n is the number vertices. First, we focus on a specific property of a neighborhood graph. We first show the following lemma.

Lemma 3. *Each element of $\mathcal{E}_{2,n}$ can be obtained in an unique way from an union of two collections of intervals of $\mathcal{E}_{1,n}$.*

Proof. Let us consider a collection $g \in \mathcal{E}_{2,n}$ and a box $i =< (x_1, y_1), (x_2, y_2) >$ in G (see Section 2.1). We give two collections of intervals I_1 et I_2 by the following way (see Fig. 6). Let $i =< (x_1, y_1), (x_2, y_2) >$ be a box in g. Then, the interval i in I_1 is (x_1, y_1) and the interval i in I_2 is (x_2, y_2).

Table 3: Heuristic 1: EDGE-COVERING heuristic.

INPUT:

- G *: graph of the target instance of CEMR*
- $cap : E(G) \rightarrow N$*: a weight function*
- H*: a neighborhood graph*
- p*: an assignment*
- r *: routing function of the instance*

OUTPUT:

- S *: a covering of* H
- ϵ *: the load of coverin* S

VARIABLES:

- g, g_{new} *: sets of vertices of* H
- $mark$*: boolean array of size* $|E(H)|$ *initialized to* $false$
- ϵ_{min} *: real*

ALGORITHM:

1. $S \leftarrow \emptyset$ */* Initialization of the current solution */*
2. For each $e = [u, v] \in E(H)$, Do $S \leftarrow S \cup \{\{u, v\}\}$
3. For each $\{u, v\} \in S$, Do $mark[\{u, v\}] \leftarrow false$
4. $\epsilon \leftarrow$ load of solution S */*(see Definition 3) */*

   / Modification of the current solution */*
5. While there exists an element $g \in S$ such that $mark[g] \leftarrow false$ Do
   (a) Choose randomly $g \in S$ such that $mark[g] \leftarrow false$.
   (b) $mark[g] \leftarrow true$.
   (c) $\epsilon_{min} \leftarrow \epsilon$ and $g_{new} \leftarrow \emptyset$ and $S_{min} \leftarrow S$

   / Choose of group such that the merge between* g *and* g' *gives the best solution */*
   (d) For all the elements $g' \in S$ such that $g' \neq g$ and $g' \cap g \neq \emptyset$ Do,
      i. $S' \leftarrow S - g - g' \cup MERGE(g, g')$

         where $MERGE(g, g') = \{x : x \in g \vee x \in g'\}$
      ii. For all the elements $g' \in S'$ such that $g' \subset g_{new}$ Do $S' \leftarrow S' - g'$.
      iii. $\epsilon' \leftarrow$ load of solution S'
      iv. if $\epsilon_{min} \geq \epsilon'$ then

         $\epsilon_{min} \leftarrow \epsilon'$ and $S_{min} \leftarrow S'$

         $g_{new} \leftarrow MERGE(g, g')$
   (e) EndFor
   (f) if $\epsilon_{min} \neq \epsilon$ then $S \leftarrow S_{min}$ and $\epsilon \leftarrow \epsilon_{min}$
6. EndWhile
7. return S and ϵ

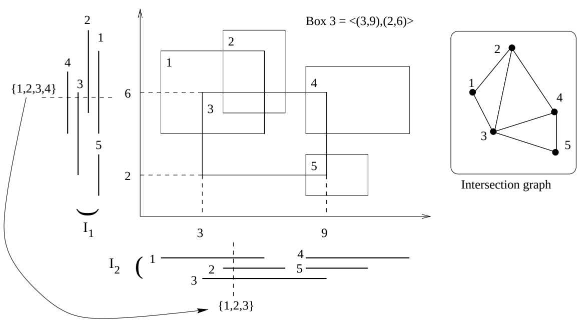


Figure 6: 2-dimensional boxes with the corresponding intersection graph and intervals

Given two intervals I_1 and I_2, we act in the opposite way to obtain g. It is clear that in every case, (g, I_1, I_2) is unique. □

Remark 1. *In a geometric way, considering that a collection $g \in \mathcal{E}_{2,n}$ is located in a positive discrete plan, the procedure given in the previous proof consists in projecting g on the two axes.*

In fact, Lemma 3 induces a bijection between $\mathcal{E}_{2,n}$ and $\mathcal{E}_{1,n} \times \mathcal{E}_{1,n}$ (see Fig. 5). For each $d \geq 2$, we can give a bijection (computed in polynomial time) between $\mathcal{E}_{d,n}$ and $\mathcal{E}_{d_1,n} \times \mathcal{E}_{d_2,n}$, for any $d_1 + d_2 = d$.

Corollary 3. *Each neighborhood graph is the intersection of an unique pair of interval graphs with n vertices sharing the same vertex set.*

Now, we will give the main result used by our heuristic.

Proposition 3. *The enumeration of the maximal induced cliques of a neighborhood graph with n vertices can be done in time $O(n^4)$.*

Proof. By Lemma 3 each collection $g \in \mathcal{E}_{2,n}$ can be associated in an unique way to a pair of collections of intervals (I_1, I_2). Let H, G_1 and G_2 be respectively the intersections graphs of g, I_1, and I_2. By construction, these three graphs have the same vertex set $V = \{1, \ldots, n\}$.

In Fig. 5, $\{1,2,3,4\}$ and $\{1,2,3\}$ are maximal clique respectively in I_1 and in I_2 and we can also notice that g has maximal clique $\{1,2,3\}$. Property 2 explains the previous remark and it is a direct consequence of the construction of I_1 and I_2 from g.

Property 2. *A subset of vertices $V' \subset V$ induces a maximal clique in H if and only if V' induces a maximal clique in a subgraph of G_2 induced from the vertices of a maximal clique in G_1.*

From a geometric point of view, let us consider the projection of I_1 (resp. I_2) on a segment s_1 (resp. s_2) (see Remark 1 and Fig. 5). From these projections, each segment is composed of at most $k_1 \leq 2n$ (resp. $k_2 \leq 2n$) consecutive sub-segments (see Fig. 5). The algorithm we give acts as follows: assume $s_1 = s_{1,1}, \ldots, s_{1,k_1}$ and $s_2 = s_{2,1}, \ldots, s_{2,k_2}$. For each sub-segment $s_{1,j}$, we define $c_{1,j}$ as the collection of the interval labels of I_1 covering this segment. It is clear that $c_{1,j}$ induces a clique in G_1. We denote by $C_{I_1} = \{c_{1,1}, \ldots, c_{1,k_1}\}$ the set of collections we obtain (in polynomial time). Consider now $\mathcal{C}_1$ a set of collections of intervals initially empty. For each $c_{1,i}$,

- we consider $I_{2,i}$ the subset of intervals of I_2 induced by $c_{1,i}$,
- we build $C_{I_{2,i}} = \{c'_{2,1}, \ldots, c'_{2,k_2}\}$, where $c'_{2,i}$ is the set of interval labels of $I_{2,i}$ covering segment $s_{2,i}$,
- $\mathcal{C}_1 \leftarrow \mathcal{C}_1 \cup C_{I_{2,i}}$.

So,

Property 3. $|\mathcal{C}_1| \leq 4n^2$ *and for each $c \in \mathcal{C}_1$ we have $|c| \leq n$.*

We denote by $\overline{\mathcal{C}}_1$ the closure of $\mathcal{C}_1$, i.e., $\overline{\mathcal{C}}_1$ is the biggest subset of $\mathcal{C}_1$ such that for each pair c, c' in $\overline{\mathcal{C}}_1$, $c \not\subset c'$ and $c' \not\subset c$. From Property 2, $\overline{\mathcal{C}}_1$ is obtained in polynomial time function of n and $|\overline{\mathcal{C}}_1| \leq 4n^2$.

From Property 2, the elements of $\overline{\mathcal{C}}_1$ are maximal cliques in H et by construction, $\overline{\mathcal{C}}_1$ contains all these cliques. □

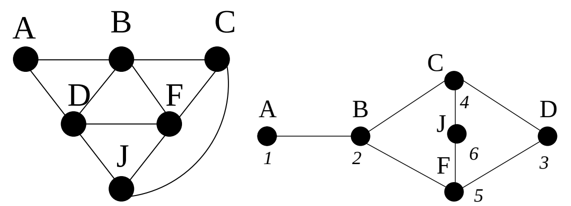
Proposition 3 leads us to obtain a covering of the neighborhood graph by maximal cliques in polynomial time $O(n^4)$. This is also the basic principle of Heuristic 2 we describe now. From the main idea given in Corollary 2, we deal here with the coverings of H in which each group induces a maximal complete graph of H. Thus, we consider a particular set of partitions of $E(H)$ for which we know by Proposition 3 that the cardinality is polynomial. In this case, Problem CEMR consists in studying another kind of problem in which we both take into account the fast construction of ad hoc coverings, and the limited load of edges in H.

Now, we will give an example to show how Heuristic 2 works (see Table 4 for a formal description). Let consider the instance of problem CEMR represented in Fig. 7. Moreover, all edges in G have the same capacity equal to 10 and the routing function r returns a steiner tree given a set of vertices and $r(\{J, D\}) = \{3, 4, 6\}$.

First, all maximal complete graphs of H are computed:

$$K = \{\{A, B, D\}, \{B, C, F\}, \{B, D, F\}, \{C, F, J\}, \{D, F, J\}\}$$

S is initialized to $\{\emptyset,\ \emptyset,\ \emptyset,\ \emptyset,\ \emptyset\}$. Set S corresponds to the solution (Note that, the number of groups of S is at most the number of maximal complete graphs of graph H). Moreover, set NC is equal to $E(H)$. Now, we focus on the process executed in the loop `While`. Graph $\mathcal{G}$ reprensents the relationship between maximal complete graphs and edges of H. An edge of graph $\mathcal{G}$ between a vertex representing a maximal clique c and a vertex corresponding in edge e of H indicates that c covers edge e in graph H. Now, the remainder of the algorithm determines how all the edges of H are covered by the solution. At each step of this process, one edge e of H is selected so that the number of groups in K containing e is minimum. Afterwards, amoung all groups covering edge e, one group is selectionned for covering edge e because this solution is the best solution for minimising the efficency. As edges $[A, B], [A, D], [B, C], [C, J], [D, J]$ are covered by one element of K, their degree in $\mathcal{G}$ is equal to 1 and they are selected first (see Fig. 8(b)). After their treatment, S is equal to $\{\{A, B, D\}, \{B, C\},\ \emptyset, \{C, J\}, \{D, J\}\}$ and $NC = \{\{B, D\}, \{B, F\}, \{C, F\}, \{D, F\}, \{D, J\}, \{F, J\}\}$ (see Fig. 8(a)). Moreover, now e is equal to edge $[C, F]$: elements $\{C, F, J\}$ and $\{B, C, F\}$ cover it. The efficiencies of covers $\{\{A, B, D\}, \{B, C\},\ \emptyset, \{C, F, J\}, \{D, J\}\}$ and $\{\{A, B, D\}, \{B, C, F\},\ \emptyset, \{C, J\}, \{D, J\}\}$ are computed (they are respectively equal to 5/10 and 4/10). Thus, S becomes equal to $\{\{A, B, D\}, \{B, C, F\},\ \emptyset, \{C, J\}, \{D, J\}\}$. And so on.



(a) a neighborhood graph H

(b) a graph G and an assignment p

Figure 7: An instance of problem CEMR

6 Simulation and experimentation results

This final section gives some simulation results about the use of the heuristics given in Section 2.2. The purpose of this section is to evaluate the quality of the solutions given by ours two solutions in compare to all valid solutions.

Since Problem CEMR is NP-complete even if G is a tree, we consider only G as the tree given in Fig. 3. Thus, the routing function r is obvious. Neighborhood graphs H are randomly chosen such that they verify the Helly property, they have the same diameter and they contain 12 edges.

For each random neighborhood graph H, the simulations we consider consists in computing the average load ϵ_1 obtained from 5 executions of H_1 and the load ϵ_2 obtained by running H_2. For the 300 generated random graphs, Fig. 9 gives the percentage of solutions found by the two heuristics. Note that they give always realizable solutions, knowing that the average ratio of realizable solutions is 31.3%, H_2 returns a solution which is in the first 17.2% of the best solu-

Table 4: Heuristic 2: CLIQUE-COVERING heuristic.

INPUT:

- *G : graph of the target instance of CEMR ; $cap : E(G) \rightarrow N$: a weight function*
- *H: a neighborhood graph ; p: an assignment ; r : routing function of the instance*

OUTPUT:

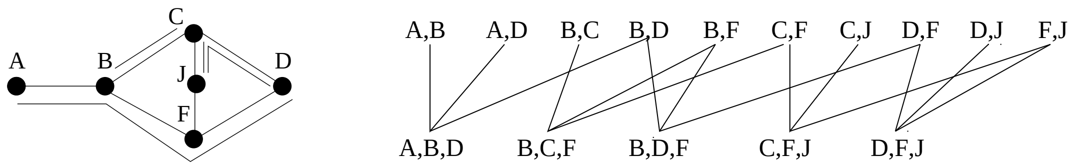
- *S : a cover of H; ϵ : the load of cover S*

VARIABLES:

- *g, g_{new} : sets of vertices of H ; ϵ_{min} : real*
- *marks: boolean array of size $|E(H)|$ initialized to $false$*

ALGORITHM:

1. Compute K where $K = \{k_1, \ldots, k_z\}$ is the set of all maximal complete graphs of H
2. $S \leftarrow \emptyset$ and $\epsilon \leftarrow 0$ /* *Initialization of the current solution* */
3. For each $i \in [1, \ldots, z]$ do , $S \leftarrow S \cup \{\emptyset\}$
4. For each $e \in E(G)$ do, $load(e) = 0$ /**Initialization of the state of network G**/
5. $NotCovered \leftarrow E(H)$/* *Initialization of computing phase* */
6. While ($NotCovered \neq \emptyset$) do /* *Phase of computing of the cover* */
   (a) Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where
   $$\mathcal{V} = NotCovered \cup K \text{ and } \mathcal{E} = \{(e, s) : e = [x, y] \in E(H) \wedge s \in K \wedge x \in s \wedge y \in s\}$$
   (b) Choose $e = [u, v] \in NotCovered$ such that $d_{\mathcal{G}}(e) = min\{d_{\mathcal{G}}(t) : t \in NotCovered\}$
   (c) $\epsilon_{min} \leftarrow \infty+$ /* *selection of element in K for covering edge e* */
   (d) For each $i \in [1, \ldots, z]$ do /**notation: $S[i]$ (resp.$K[i]$) is the ith element of S (resp.K)**/
      i. if $(u \in K[i]) \wedge (v \in K[i])$ then
         A. $g_{new} \leftarrow S[i] \cup \{u, v\}$ and $\epsilon_{new} \leftarrow \infty+$
         B. For each $e \in E(G)$ do
            /**Suppression of communication tree of group $S[i]$**/
            - if $e \in r(S[i])$ then $load'(e) \leftarrow load(e) - |S[i]|$ otherwise $load'(e) \leftarrow load(e)$
              /**Taking count of communication tree of new group g_{new}**/
            - if $e \in r(g_{new})$ then $load'(e) \leftarrow load'(e) + |g_{new}|$
            - if $\epsilon_{new} < load'(e)/cap(e)$ then $\epsilon_{new} \leftarrow load'(e)/cap(e)$
         C. EndFor
         D. if $(\epsilon_{min} > \epsilon_{new})$ then $nb \leftarrow i$ and $(\epsilon_{min} \leftarrow \epsilon_{new})$/* *keep the best solution* */
   (e) EndFor /* *update the load of network G + the cover which now covers e**/
   (f) $NotCovered \leftarrow NotCovered - \{e\}$
   (g) For each $e \in E(G)$ do, if $e \in r(S[nb])$ then $load(e) \leftarrow load(e) - |S[nb]|$
   (h) $S[nb] \leftarrow S[nb] \cup \{u, v\}$ and $\epsilon \leftarrow \epsilon_{min}$
   (i) For each $e \in E(G)$ do, if $e \in r(S[nb])$ then $load(e) \leftarrow load(e) + |S[nb]|$
7. EndWhile
8. return S and ϵ



(a) A mapping of covering $\{\{A, B, D\}, \{B, C\}, \emptyset, \{C, J\}, \{D, J\}\}$

(b) A graph $\mathcal{G}$

Figure 8: An instance of problem CEMR

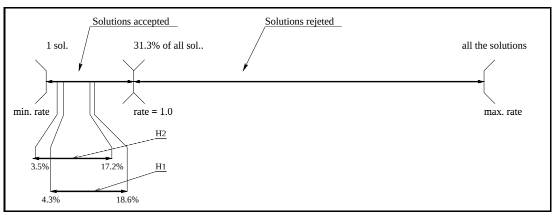


Figure 9: Percent for the acceptable solutions for Heuristics H_1 and H_2

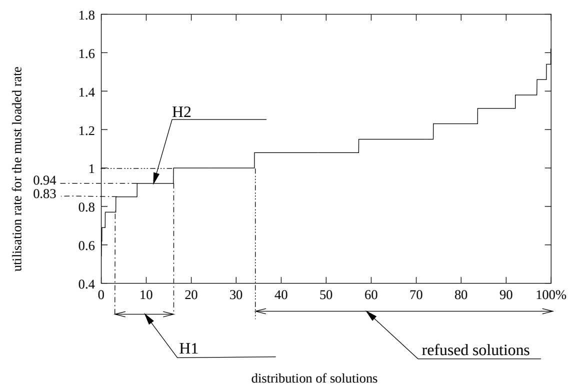


Figure 11: All solutions and the quality of Heuristics H_1 and H_2

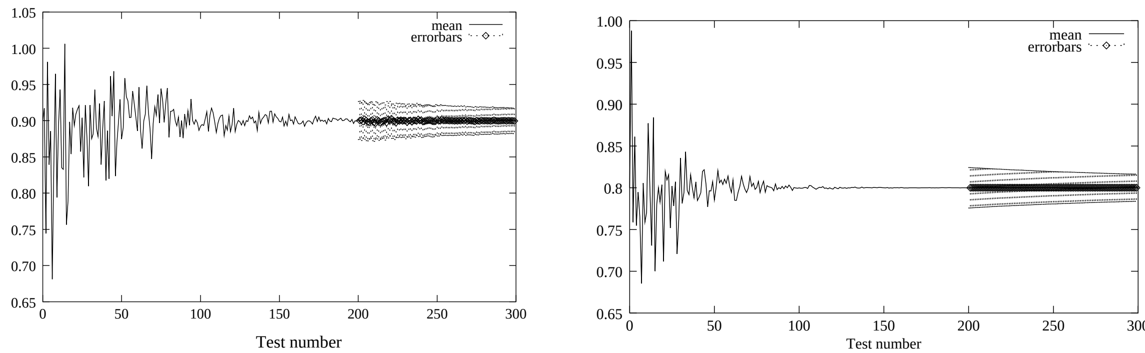
tions and H_1 returns a solution which is in the first 18.6% of the best solutions. Fig. 10 and 11 presents the average values of ϵ_1 and ϵ_2 on 300 random graphs and their standard deviations. We get (using technic described in [17]): $\overline{\epsilon_1} = 0.89^{\pm 0.09}$ and $\overline{\epsilon_2} = 0.84^{\pm 0.13}$ (see Fig. 10(a) and 10(b)). Note that here, Heuristics H_1 and H_2 provide realizable solutions.

We consider now a particular instance considering the two graphs G and H given in Fig. 3 and 2. The capacity of each edge of G is equal to 10. Fig. 11 presents the distribution of all the solutions (i.e., the efficiencies of all the possible mappings), sorted by increasing values. Note that only 34% of the mappings are realizable on G.

These simulation results, for which we analyzed the confidence, lead us to some remarks and analysis. We first see that the random aspect of H_1 can be used to obtain good solutions by making many executions of it on the target instance. The average behavior of H_2 is better than the one of H_1 because H_2 considers the topology of the neighborhood graphs. Actually, we also considered regular topologies of graphs H, that do not contain large cliques: H_1 and H_2 turn out to return equivalent solutions. An open question is to determine what are the realistic properties of neighborhood graphs in addition to the Helly property we could use to improve H_2.

7 Conclusion

We investigate here a theoretical problem related to communication resource allocation to answer multicast requirements for distributed interactive simulations. Our approach consists in considering it as a multipoint communication problem for which



(a) Mean of parameter ϵ_1 and its confidence interval

(b) Mean of parameter ϵ_2 and its confidence interval

Figure 10:

we give some complexity results (NP-complete problems), and some lower bounds. We also developped two polynomial-time heuristics. They provide anwers that are experimentally shown to be in the top 19% of the best solutions. There remain (at least) three open questions about this problem. First, we have seen that neighborhood graphs have the Helly property (we use in Heuristic 2), but is it the only main characteristic of these graphs in practice? Second, could we find some algorithm with approximation guarantees to solve Problem CEMR for some classes of network graphs (for example trees) and/or of neighborhood graphs? Finally, Problem CEMR can be seen as the initial problem to be solved at the beginning of the simulation. When the neighborhood graphs change time after time, the initial solution has to be quickly modified. What is the ad hoc algorithmic approach to be use to do this modification? For which kinds of changes in the neighborhood graphs this modification of the multipoint solution has to be done? Some tests using local search technics can be founded in [11].

ACKNOWLEDGEMENTS

The work reported in this paper has been partially supported by a grant from DGA and SYSECA (THALES).

REFERENCES

1 M.H. Ammar, G.C. Polyzos, and S.K. Tripathi. Network support for multipoint communication. *Special Issue of IEEE Journal on Selected Areas in Communications*, 15(3):273–276, 1997.

2 E. Asplund and B. Grunbaum. On a coloring problem. *Math. Scand*, 8:181–188, 1960.

3 E. Balas and C.S. Yu. On graphs with polynomially solvable maximum-weigth clique problem. *Network*, 19:247–253, 1989.

4 A.J. Ballardie, P.F. Francis, and J. Crowcroft. Core based trees (cbt). In *ACM SIGCOMM*, pages 85–95, 1993.

5 D. Barth, P. Berthomé, and P. Fragopoulou. Heuristics for the simultaneous satisfaction of multipoint requests on networks. Technical report, Université de Paris-Sud, 2001.

6 S. Bellantoni, I. B.-A. Hartman, T. Przytcka, and S. Whitesides. Grid intersection graphs and boxicity. *Discrete Math.*, 114(41–49), 1993.

7 C. Berge. *Graphs and Hypergraphs.* North-Holland Mathematical Library, 1885.

8 H. Bettahar and A. Bouabdallah. QoS routing protocol for the generalized multicast routing problem (GMRP). In *EEE International Conference on Networking, ICN'01*, volume 2093 of *Lecture Notes in Computer Science*, pages 630–638, 2001.

9 A. Bielecki. Problems 56. *Colloq. Math*, 1:333, 1948.

10 J. Crowcroft and M. Hofmann. Networked group communication. In *Third International COST264 Workshop, NGC*, 2001.

11 C. Durbach. *Optimisation des communications dans des simulations distribuées*. PhD thesis, Université de Versailles, 2001.

12 D. Estrin et al. Protocol independent multicast-sparse mode (pim-sm): protocol specification.a proposal to add explicit congestion. RFC 2117, June 19991997.

13 R. Fujimoto. *Parallel and Distributed Simulation Systems*. Wiley-Interscience, 2000.

14 Michael Garey and David Johnsom. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. A series of books in the Mathematical sciences. W.H. Freeman and Co., 1979.

15 P. Hanlon. Counting interval graphs. *Journ. of Trans. Am. Math. Soc.*, 272:383–426, 1982.

16 A. Irlande, J.-C. König, and C. Laforest. Construction of low-cost and low-diameter steiner trees for multipoint groups. In *nternational colloquium on structural information and communication complexity (SIROCCO)*, pages 197–210, 2000.

17 R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. J. Wiley and sons, 1991.

18 C. Kanarick. A technical overview and history of the simnet project. *Advances in Parallel and Distributed Simulation*, 23:104–111, 1991.

19 J. Kratochvil. A special planar satisfiability problem and a consequence of its np-completeness. *Discr. Appl. Math*, 52:233–252, 1994.

20 J. Kratochvil and J. Nesetril. Indepedent set and clique problems in intersection-defined classes of graphs. *Journ. Commentat. Math. Univ. Carol.*, 1(31):85–93, 1990.

21 F.T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with application to approximation algorithms. In 29^{th} *IEEE FOCS*, pages 422–431, 1988.

22 E. Levy, T. Turletti, and F. Baccelli. Cell-based multicast grouping in large-scale virtual environments (poster session) (extended abstract). In *Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 108–109. ACM Press, 2000.

23 Z. Li and P. Mohapatra. QMBF: A QoS-aware multicast routing protocol. *Computer Communications Journal*, 2003.

24 M. Macrdonia, D. Pratt M. Zyda, and P. Brutzman. Exploiting reality with multicast groups: a network architecture for large scale virtual environments. In *IEEE Virt. Real. Annual Symp.*, pages 11–15, 1995.

25 T.W. Mastaglio and R. Callahan. A large-scale complex environment for team training. *IEEE Computer*, 28(49–56), 1995.

26 N.F. Mir. A survey of data multicast techniques, architectures, and algorithms. *IEEE Communications*, 39(9), 2001.

27 K. Morse. nterest management in large scale distributed simulation. Technical Report TR96-27, Univ. of California, 1996.

28 K. L. Morse and J. S. Steinman. Data distribution management in the hla multidimensional regions and physically correct filtering. In *15th Workshop for the Interopreability of Defense Simulation*, 1997.

29 E.T. Powel, L. Mellon, J.F. Watson, and G.H. Tarbox. Joint precision strike demonstration (jpsd) simulation architecture. In 14^{th} *Work. SIDS'96*, pages 807–810, 1996.

30 E. Prisner. A journey through intersection graph county, 2000.

31 R. Saad. Complexity of the forwarding index problem. *IAM Journ. of Discrete Math.*, 6, 1993.

32 E.S. Scheinerman. Characterizing intersection classes of graphs. *Discrete Math*, 55:185–193, 1985.

33 I. Tacic and R.M. Fujimoto. Synchronized data distribution management in distributed simulations. In *IEEE PADS*, pages 108–115, 1998.

34 D.G. Thaler and C.V. Ravishankar. Distributed center-location algorithms. *IEEE Journal on Selected Areas in Communications*, 15(3):291–303, 1997.