Multicast Routing Problem in Communication Networks: State of the Art

Nadia Saber[1] and Mohammed Mestari[2]
[1] LPRI Laboratory, Moroccan School of Engineering Sciences EMSI Casablanca, Morocco
[2] SSDIA laboratory,ENSET Mohammedia, Morocco
Email: nadiasaber4@gmail.com; mestari@enset-media.ac.ma

Abstract —Multicast routing in communication networks consists of transmitting data from one (or more) sources to multiple destinations, while minimizing the use of network resources and taking into account multiple QoS (quality of service) parameters. In an effort to understand the benefits and drawbacks of existing methods, this state-of-the-art paperaims to provide a tutorial that is easy to follow for readers who are not already familiar with the subject, and make a comprehensive survey and comparisons of different techniques employed in or proposed for routing problem.

Index Terms—Multicast routing, optimization problem, neural networks, QoS.

I. Introduction

In recent years multicast routing has become more and more popular, and has motivated several researchers, because of the progress in the area of multimedia communications, file sharing, interactive games, videoconferencing, on-demand video, radio and TV transmission... etc

Multicast routing is the ability of transmitting simultaneously a message from one or more sources to a set of appropriate destinations in a communication network; It consists of determining an optimal diffusion of packets according to certain performance criteria. It deals with constructing paths with minimal consumption of resources (bandwidth, delay, cost...).

This problem is known in graph theory as the Steiner tree problem, and has been shown to be NP-complete (non deterministic polynomial-time complete) ([1] and [2]).

Several methods are proposed in the literature to solve the Steiner tree problem. [3] and [4] proposed exact algorithms to solve this problem, but they are not viable in very large networks, because of their high degree of computational complexity. Heuristic proposed in [5] is one of the famous methods used to solve this problem, because they construct a feasible solution within reasonable time. This method assumes that the source node can obtain topology information about the communication network through the routing protocol, but it suffers from some drawbacks such as failure on the central node, or high communication cost in keeping network information up-to-date, especially in large networks. To overcome this, [6] and [7] proposed distributed algorithms for the routing problem, where each node operates based on its local routing information and the coordination with other nodes is done via network message passing.

The QoS-constrained multicast routing problem include routing that guarantees the required quality of service (QoS), such as bandwidth requirement, delay constraint on transmitting information between a source node and each destination. Many metaheuristics are proposed in the literature to solve the QoS-constrained multicast routing problem: genetic algorithms ([8]-[10]), taboo search ([11]-[13]), ant colony optimization ([14]-[16]), fuzzy-based algorithms ([17], [18]).

To solve the multicast routing problem, neural networks are also proposed. They were first proposed in [19], by defining proper energy functions and deriving associated weights between neurons. In 1982, and after the proposition of the Hopfield neural network [20], many researchers have been exploring HHNs and improving their performance on different real time applications. [21] proposed a modification of HNNs to solve constrained multicast routing, but [22] demonstrated that they are not efficient in large networks.
The multicast routing problem can be formulated as a single-objective problem (SOP) where only one generic cost function is considered ([5]-[8], [23]), or as a multi-objective problem (MOP), where several objective functions may be optimized (minimized or maximized) in conflicting situations ([24]-[27]). The proposed method to solve this problem, constructs an optimal multicast tree under several constraints, and don't need any central node to keep information about the whole network.

The present work is organized as follows: in the section 2, we will describe the mathematical modeling of a routing problem. In section 3, a description of the routing theory is given. We briefly present the quality of service in a communication network in section 4. Section 5 is dedicated to presenting the different approaches proposed in the literature to solve the routing problem. Finally, in section 6 we give a conclusion.

II. Mathematical Modeling of the Routing Problem

This section introduces the mathematicalmodeling of the routing problem, as well as all the related notions.

Manuscript received August 25, 2018; revised April 1, 2019.
doi:10.12720/jcm.14.5.408-422

A. Physical Network

A communication network is usually modeled by a graph. A graph consists of a set of vertices V and a set of edges E. An edge connecting vertex u to vertex v is represented by (u, v). The vertices of the graph represent the nodes of the network, and the edges represent the communication links between nodes.

Links in communication networks have several properties that are represented by the weights associated with the corresponding edges in the graph.

There are two types of communication links: symmetrical and asymmetrical links. Symmetrical links have the same weight in both directions, where asymmetrical links have different weights for different directions. If all the links in a communication network are symmetrical, then this network can be modeled by a non-oriented graph. In this type of graphs, the direction of the link does not matter: a link between vertices u and v can be represented by (u, v) or (v, u). Generally, communication networks are modeled by non-oriented graphs. In the rest of this work, the term "graph" will always refer to a non-oriented graph.

We denote by $G = (V, E)$ the graph modeling the communication network, where $V = \{v_1, v_2, \dots, v_N\}$ represents the set of vertices and E is the set of edges:

$E \subseteq \{(v_i, v_j)/v_i \in V, v_j \in V,$ where $v_i \neq v_j, 1 \leq i \leq N$ and $1 \leq j \leq N\}$.

Nrepresents the number of nodes in the network. i.e. $|V| = N$.

We denote by s the source vertex, u a destination vertex and $U = \{u_1, u_2, \dots, u_N\}$ the set of destination nodes. $M = s \cup U$is the multicast group.

The routing problem is known, in the graph theory, as the Steiner tree problem, and is defined as follows:

- A non-oriented graph $G = (V, E)$;
- A cost function, which associates to each edge(u, v) a positive real number c_{uv};
- A multicastgroup $M \subset V$.

We attempt to get a tree $T = (V_T, E_T)$ that covers the set M, such as the cost $c_T = \sum_{(u,v)\in T} c_{uv}$ is minimized. This minimal tree is called the Steiner tree.

Steiner's problem is an NP-hard problem; however there are trivial cases where it can be solved in polynomial time:

- $|M| = 2$ (unicast routing): there are only 2 nodes in the multicast group: the problem turns into the shortest path problem.
- $|M| = |V|$ (broadcast routing): in this case, the multicast group includes all nodes in the network: the problem becomes a problem of minimal spanning tree.
- Gis a tree: in this case, there is only one sub-tree that covers the multicast group. This sub-tree represents the solution of Steiner's problem.

A path in a graph is a finite sequence vertices $v_1, v_2, \dots, v_p$ in the manner that for each $1 \leq k \leq p$, $\{v_k, v_{k+1}\} \in E$ and $v_k \in V$. A cycle is a path $v_1, v_2, \dots, v_p$ where$v_1 = v_p$.

A graph G is connected if for any two vertices v_i and $v_j$in V, there is a path whose endpoints are v_i and v_j. A spanning tree of a graph G is a connected, acyclic and spanning subgraph which is a tree. A multicast tree is a spanning subgraph of G that includes all the vertices of M.

B. Adjacency Matrix and Decision Variables

We assume that the graph has N vertices given by $V = \{v_1, v_2, \dots, v_N\}$. The adjacency matrix $X = (x_{ij})_{\substack{1\leq i\leq N \\ 1\leq j\leq N}}$is a square boolean matrix defined by:

$$x_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The final solution is given by the matrix $Y = (y_{ij})_{\substack{1\leq i\leq N \\ 1\leq j\leq N}}$ defined as follows:

$$y_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in T \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Fig. 1 illustrates an example of a network, modeled by a graph G with 8 vertices, the source vertex is $s = \{8\}$ and the set of destination vertices is $U = \{1,2,3,5\}$ .The multicast group is then: $M = \{1,2,3,5,8\}$.

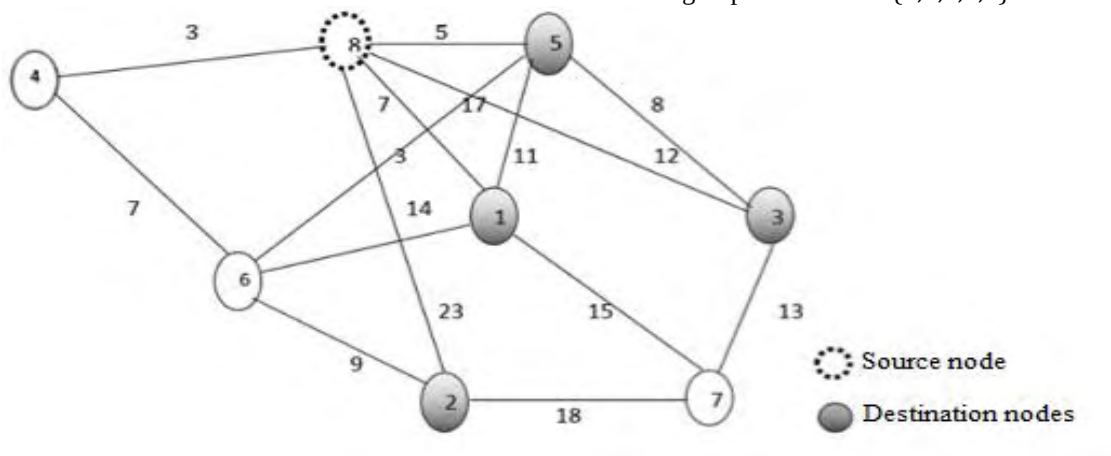

Fig. 1. A weighted graph G modeling a network with 8 nodes

The adjacency matrix associated to the graph of Fig.1 is defined as follows:

$$X = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

There is no exact algorithm that constructs a Steiner tree in a polynomial time; therefore it is necessary to develop approximate algorithms to solve this problem.

Approximate algorithms for the Steiner tree problem can be performed in polynomial time and give "good" solutions (not necessarily optimal). Most of them are based on finding shortest paths to connect different components of the tree.

III. The Routing Theory

Depending on the number of destination nodes, there are four types of routing:

- Unicast routing (one to one)
- Anycast routing
- Broadcast routing (one to all)
- Multicast (one to many)

A. Unicast Routing

In a communication network, unicast routing (Fig.2 (a)) is used to switch messages between two entities. A source entity having data to be transmitted to a well-defined destination.

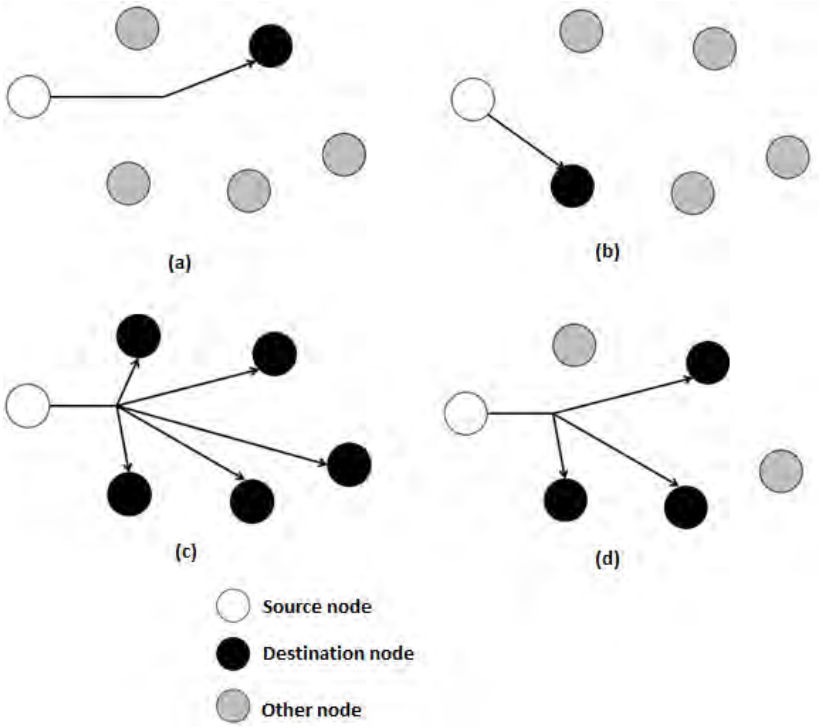

Fig. 2. Different types of routing

This communication can be direct, i.e. the destination is directly accessible from the source, and messages are then directly sent to this destination, or indirectly and performed in multi-hops. Unicast communication to an out-of-reach destination usually involves the research of a path from the source to the destination through other nodes in the network. A path is then constructed between these two entities.

B. Anycat Routing

Anycast routing (Fig.2 (b)) in a communication network consists of delivering a message to a single member of a network, usually the closest. Anycast is a routing technique that switches data to the "nearest" or "most efficient" server in line with the routing protocol.

C. Broadcast Routing

The broadcast/multicast routing was introduced by Deering in the early 90's. He found that sending a single message from a source to each node requires a lot of resources (bandwidth). He then proposed the sending of a single message by the source, and its duplication during the routing [28]. Broadcast routing (Fig.2 (c)) refers to the ability of a communication network to deliver a single message simultaneously to all connected machines in the network.

D. Multicast Routing

1) Definitions

Multicast routing (Fig.2 (d)) is the ability of a communication network to simultaneously deliver a

single message to multiple destinations, located in different geographical positions. One of the major challenges is to minimize the use of network resources used by multicast routing.

A multicast group can be static or dynamic. Static groups cannot be changed once created, while in dynamic groups nodes can join in or leave out the group at any time [29].

Multicast groups can also be classified according to the relative number of participants [28]. In "sparse groups", the number of participants is small compared with the number of nodes in the network. In the other situation, in which most network nodes are participating in multicast communication, the groups involved are called "pervasive groups".

Today, many applications require this type of routing, such as file sharing, interactive games and video conferencing. Multicast applications are becoming increasingly popular and greedy of resources, resulting a pressing need to develop more powerful tools to support these applications.

In general, different multicast applications have different requirements, that is why multicast communications fall into two categories:

- SSM (source-specific multicast): it is a form of a communication where only one node of the multicast group sends data, while others receive it.
- ASM(any source multicast): (shared group): it is a type of multicast routing where each member of the multicast group can exchange (send or receive) data with other members of the group.

2) Multicast routing algorithms

Let $G = (V, E)$ be an undirected graph where V is the set of vertices and E is the set of edges. Recall that if G is undirected, it models a communication network with symmetrical links. Let M be the multicast group. The routing problem in communication networks is equivalent to finding a tree T in the graph G that spans$M$. The latter is called Steiner tree or multicast tree.

Like multicast communications, multicast trees are also classified into two categories: source-based trees (corresponding to the source-specific multicast) and shared trees (corresponding to ASMs).

In what follows, we will see the properties of a good multicast tree.

For the majority of multicast applications, some properties are more significant compared with the others.

For that, we divided these properties according to their priority level: high, medium and low.

• High priority level:

a) Low Cost: the cost (or weight) of a multicast tree is the sum of the costs (or weights) of all edges in this tree. An optimal multicast tree has minimal cost.

b) Minimum delay: the transmission delay from a source to a destination is the sum of all transmission delays of edges involved in this transmission. Optimal trees minimize transmission delays for each source-destination couple in the multicast groups.

c) Scalability: a good multicast tree is scalable in two ways. First, the construction of a multicast tree for large networks must be done in a reasonable time. Second, commutators in communication networks must be able to support large numbers of multicast trees simultaneously.

• Medium priority level:

a) Supporting dynamic multicast groups: a good multicast tree should allow members to join in or leave out the tree without being affected.

b) Survivability: a good multicast tree must survive without being affected by nodes or edges failure.

• Low priority level:

Impartiality: a good multicast tree is impartial in two aspects. First, it attempts to provide a minimum quality of service (e.g. delay) to each member of the multicast group. (It is unfair to unnecessarily punish a member to improve the quality of service of other members of the group). It attempts to evenly distribute multicast effort (i.e. the packet duplication effort) between all participating nodes.

IV. Quality of Service: QoS

First, it is important to specify what the term "routing with quality of service" covers. Quality of Service (QoS) is the performance criteria that networks require to have in order to satisfy users' needs. These performances correspond to different measurable parameters on the networks such as ([30]) :

- End-to-end delay: time taken to transfer a packet between two nodes;
- Jitter: variation of time interval between two packets during their routing between source and destination;
- Bandwidth: total amount of information that a link between two nodes can absorb without creating a queue;
- Packet loss rate: the number of packets lost compared to the number of packets delivered.

Depending on the applications considered, the parameter to be taken into account differs: for example, looking for a path with a certain amount of bandwidth for video traffic or a path ensuring that the packets will be received by the destination in less than a certain period of time after they are delivered by the source. Any path satisfying a certain quantitative criterion can be qualified as a path ensuring a certain quality of service. On the other hand, each application has specific needs and only the application is aware of its needs. It is difficult to define overall service levels. Applications must be able to explicitly indicate, for example, the maximal delay that they can support.

V. Methods for Solving the Multicast Routing Problem in Communication Networks

A. Exact Algorithms

The exact algorithm proposed by Chow [3] is based on the dynamic programming technique proposed by Dreyfus and Wagner [31].

This technique starts with finding all optimal multicast trees for small subsets of D (D is the set of destination vertices). Next, it determines all optimal multicast trees for larger subsets ofD, by merging trees found previously.

This algorithm has several drawbacks; the first one is that it only minimizes a single objective (cost) function without taking into consideration other constraints. The second is the high complexity in the worst case: in a network with 20 nodes, where 5 nodes represent the destination set, the algorithm must run 12,700 times [3]. This number increases exponentially with the size of the destination set.

Actually, there is no algorithm constructing an exact minimal Steiner tree in polynomial time. However, several heuristics trying to reduce the complexity of this problem have been proposed. In the next section, we will present some heuristics to approach the optimal solution.

B. Heuristics

Several heuristics approaching the optimal solution of the problem of multicast routing in a reasonable time have been proposed in the literature. Many of these heuristics resume the basic ideas for building minimal spanning trees (whose complexity is easily controllable) or shortest paths between vertices to build a multicast tree relatively advantageous.

1) KMB's heuristic

KMB's heuristic, proposed by Kou, Markowsky and Berman [32] is a method that mimics the basic idea of Prim's algorithm [33] for minimum spanning tree. This heuristic considers the complete graph G_1 of de graph G which models the communication network. A complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge. This edge represents the shortest path between these vertices. The steps of the algorithm are:

1. Construct the complete graph G_1 from the initial topologyG.

2. Find the minimum spanning tree T_1 of G_1 (if more than one minimum spanning tree exists, choose one arbitrarily).

3. Construct a subgraphG_S of G by replacing each edge in T_1 by the shortest path corresponding in G (if more than one path exists, choose one arbitrarily).

4. Find the minimum spanning tree T_S of G_S (if more than one minimum spanning tree exists, choose one arbitrarily).

5. Construct the T_H multicast tree from T_S , by removing, if necessary, edges from T_S in the manner that all leaves are multicast nodes.

2) Kompella's heuristic

The proposed method by Kompella*et al.* [5] is one of the best known methods for solving this problem because it builds a feasible solution within a reasonable time. This algorithm assumes that the source has all the information concerning the communication network to build the multicast tree, and this via the routing protocol. The obtained multicast tree is a minimal cost tree, under delay constraint.

In this method, authors look for the shortest paths connecting all pairs of vertices. Therefore, the transitive closure G′ of G can be constructed. The next step is to build a tree that spans G′. The authors use a greedy approach to add edges to the spanning tree until all destination vertices are included.

This method builds a tree relatively advantageous within a reasonable time, but it has several drawbacks such as the failure of the central node of the network, or the necessary cost to update the network information, especially in large networks.

To remedy this problem, distributed algorithms have been proposed for the multicast routing problem, where each node of the communication network acts based only on its local routing table, and coordination between the different nodes is provided via the routing protocol.

C. Distributed Algorithms

a) Bauer and Varma's algorithms

Bauer and Varma [6] proposed two distributed algorithms.The first one is a distributed implementation of the Takahashi and Matsuyama's algorithm [34]. This construction is limited to the multicast group with the following steps listed below:

- Initialization: initialize the tree by choosing an arbitrary vertex in the multicast group and then calculate distances between this vertex and al the other members of the group.
- Reaching the last vertex in the group: add to the current tree the nearest vertex (using the shortest path between this vertex and the tree). Calculate distances between the remaining vertices and the resulting tree.

The second method represents the distributed implementation of the algorithm proposed by Kruskal [35]. This version considers all members of the multicast group simultaneously. It proposes to initialize a forest (a set of trees) with the members of the group (each vertex initializes an isolated tree). This algorithm allows obtainingof a single spanning tree by successively connecting the isolated trees. In this algorithm, connections are made using the shortest paths:

- Initialization: we consider each member of the group as an initial isolated tree.
- Reaching the obtaining a single tree: calculate distances between trees and then connect the two nearest trees using the shortest path between them.

The distributed version of this algorithm is similar to the centralized version. The trees of the initial forest are called components. At each iteration, each vertex of the graph is either a part of a component or is not yet included in the multicast tree. Each component has a leader which coordinates these activities. It executes the finite state machine represented by the Fig. 3. At each iteration, each component tries to join the nearest component and this is done in two steps:

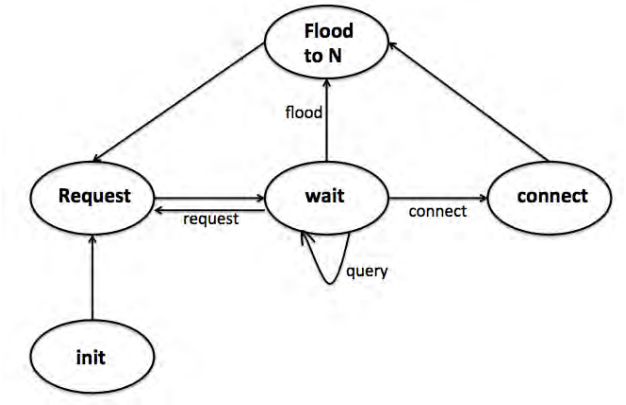

Fig. 3.A Finite state machine of the Bauer and Varma's distributed algorithm [6]

1. Discovery: In this step, the leader vertex collects and updates information about the other components and vertices of the graph.

2. The connection:In this step, the leader vertex communicates with the nearest component and requests a merge. If the request is accepted, one of the two leaders connects the two components.

The cycle is repeated until the end of the algorithm regardless of the response to the merge request.

The finite state machine:

- "init" state: at the beginning of the algorithm, each multicast vertex is the leader of the component formed by itself, and knows beforehand the distance that separates it from all other components. At this point, the discovery step is not necessary. Thus this algorithm starts directly with the connection step.
- The states "request, wait and connect": these states constitute the connection step. The leader sends a merge request to the nearest component and in this case we have three possible answers: accepted, rejected or busy.
    - Busy: a component leader sends this response when the request is made during the discovery step. After receiving this response, the component must re-request the merge.
    - Rejected: a vertex sends this answer if:
        - The component receives a connection request from another component;
        - The connection is not allowed;
        - It is no longer a leader of the component;
    - Accepted: This response is returned when the merge requests are mutual.

The second algorithm is a special case of the first algorithm described above. The only difference is that each multicast vertex can play the role of source node, unlike the first algorithm when only the source component can connect the other components to itself.

b) Jia's algorithm

Jia [7] presents another algorithm to solve the delay constrained routing problem. It assumes that the minimum cost path between two vertices is the minimum delay path, which is not always true.

To facilitate the implementation, the algorithm uses a table to keep track of the following information regarding a destinationu_i:

- The vertex is currently included in the tree or not;
- The path cost separating u_i from the tree;
- The closest vertex to u_i.

The basic idea of this algorithm is practically the same as that of the Takahashi and Matsuyama's algorithm [34], where in each iteration, the closest multicast vertex to the tree joins it (by using the shortest path between this vertex and the tree), and which respects the delay constraint.

There are two important criteria for evaluating a distributed routing algorithm: the number of messages and the convergence time.

In the worst case, the complexity of Jia's algorithm is $\Theta(2m)$, where m is the number of destination vertices [7]. For the Bauer and Varma's algorithm, the worst-case complexity is $\Theta(mn)$, where n is the size of the network.

D. Metaheuristics

Several meta-heuristics are proposed in the literature for the problem of constrained multicast routing problem. These meta-heuristics aim to approach a feasible solution in a reasonable time. Among these meta-heuristics, we can mention:

- Genetic algorithms;
- Taboo search;
- Ant colony algorithms;
- Neural networks;

a) Genetic algorithms

Genetic algorithms [36] are based on the principles of natural genetics and biological evolution.

By defining an appropriate maximum number of iterations to the genetic algorithms, an approximation of the optimal solution can be obtained in a reasonable time. In these algorithms, candidate solutions are encoded as chromosomes. Furthermore, the idea of natural selection is applied, such as selection and mutation operations, to obtain better chromosomes.

Genetic algorithms are typically implemented as follows:

- **Step 1**: initialize a population of chromosomes (solutions);
- **Step 2**: evaluating each chromosome in the population;
- **Step 3**: create new chromosomes by crossing existing chromosomes, and applying the mutation principle;
- **Step 4**: remove elements from the population which are replaced by new chromosomes;
- **Step 5**: evaluate the new chromosomes to be inserted into the population;
- **Step 6**: when a stopping criterion is satisfied, then terminate the algorithm and return the best chromosome, otherwise go back to step 3.

The GAMRA algorithm

The $GAMRA$ genetic algorithm proposed by Hwang et al. [37] to solve the delay constrained routing problem starts by constructing acceptable delay paths and stores them in a routing table.

For a source node s and a set of destination nodes $D = \{u_1, u_2, \ldots, u_k\}$, a chromosome can be represented by a sequence of integers of length k. A gene g_i of the chromosome, where $1 \leq i \leq k$ is an integer in $\{0,1,\ldots,R-1\}$which represents a possible path between s and u_i where $u_i \in D$, R is the number of paths registered in the routing table of node u_i. The relation between gene, chromosome and routing table is illustrated byFig. 4

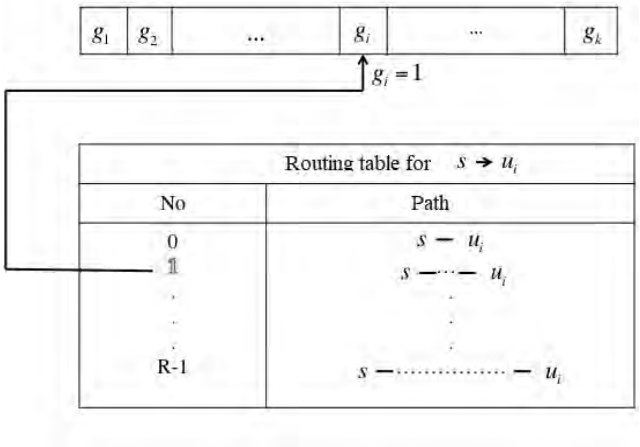

Fig. 4. Relation between gene, chromosome and routing table

A chromosome represents a candidate solution for the multicast routing problem because it guarantees a path between the source and each destination vertex.

Genetic algorithms initialize a population of chromosomes, each of them has a fitness value that defines the quality of the chromosome. The different steps are as follow:

1. Initialization of the population: the first step is to randomly generate P different chromosomes.

2. Chromosome evaluation: the fitness value of a chromosome corresponds to the value of the objective function represented by this chromosome. This value is calculated as follows:

$$F(h_i) = 1 - \frac{C(h_i)}{C(L)} \quad (3)$$

where:

- h_i: represents a chromosome;
- $C(h_i)$: represents the cost of the graph encoded by h_i;
- $C(L)$: represents the total cost of the graph representing the initial topology.

Then, the chromosomes will be sorted in descending order of their fitness values such as:

$$F(h_0) \geq F(h_1) \geq \ldots \geq F(h_{P-1}) \quad (4)$$

The first chromosome represents the best found solution.

3. Remove duplicated chromosomes:The application of genetic operations such as crossing, on 2 duplicated chromosomes will generate the same chromosome. Therefore, duplicated chromosomes will be replaced by new randomly generated chromosomes.

4. Selection: for this process, a number of chromosomes of best fitness values in the current population will be selected.

5. crossing: during this operation, two chromosomes of important fitness values exchange parts of their chains. The starting point and the length of the portion to be exchanged are randomly selected. Two new offspring are created and inserted into the population.

6. Mutation: this process is a kind of change within a chromosome. In this approach, a bit of the chromosome chain is changed with a certain probability, defined as the mutation rate.

The drawbacks of this approach are the time required to build the routing tables, and the standard crossover procedure, which gives a non-robust algorithm that converges slowly.

Zhengying*et al*.

The genetic algorithm proposed in [8] adds the bandwidth constraint to the previous problem.

In this approach, a chromosome represents a multicast tree. Before the population initialization step, a refining operation is envisaged. During this operation, the edges that have a lower bandwidth compared with the one required by the routing will be removed. Therefore the remaining edges in the refined graph must satisfy the bandwidth constraint.

The authors use a depth-first search algorithm to construct random multicast trees, which will form the initial population. Trees thus built do not necessarily respect the delay constraint.

The fitness function is defined as follows:

$$\mathrm{f(T)} = \frac{\alpha}{\sum_{e \in T} \mathrm{cost(e)}} \prod_{t \in D} \Phi(\mathrm{delay}(P_T(s,t)) - \mathrm{DDFC(t)}) \quad (5)$$

$$\Phi(Z) = \begin{cases} 1 & Z \leq 0 \\ 0 & Z > 0 \end{cases}$$

where:

- α : Apositive real coefficient;
- Φ(Z): Apenalty function;
- DDFC:Thedestination delay function constraint. This function assigns an upper bound of delay between the source and the destination t;
- $\text{delay}(P_T(s,t))$: The sum of the edge delays which constitute the path between the source s and the destinationt.

If the path satisfies the delay constraint (i.e. $\text{delay}(P_T(s,t)) \leq \text{DDFC}(t) \quad \forall t \in D$), the value of Φ is 1, otherwise is $r(0 < r < 1)$. The value of rdetermines the degree of penalty.

1. Selection: at this stage, optimal individuals are first chosen and directly copied into the next generation, then the other individuals are chosen according to a model.

The probability sp_i of selection of the parent i is given by the following formula:

$$sp_i = \frac{f(T_i)}{\sum_{i=1}^{NP} f(T_i)} \quad (6)$$

2. Crossover: the Crossoverprocessisrepeated $N_P - N_{best}$ times, where N_P is the size of the population and N_{best} is the number of the best individuals that are directly included in the next generation.

Duringthisprocess, two parent trees are selected to produce a childtree. Wechoose the commonedges in the two parent trees and copy themdirectly to the childtree. Thesesamecommonedgescanform, in some cases, separatesubtrees, to whichwecanaddotheredges to transformtheminto a multicast tree.

When connecting separate subtrees, we first select two random subtrees and connectthem via the minimum cost or minimum delay path. If no parent satisfies the delay constraint, we choose the minimum delay. The new subtree thus constructed will be included in the next selection. Therefore, this selection process is repeated until the construction of a multicast tree is completed.

3. Mutation: The mutation procedure chooses randomly, according to a probability p_m, a set of vertices, and splits the multicast tree in to subtrees by deleting all the edges that have one of the selected vertices as endpoints. Then, the sesubtrees are connected by minimum cost or minimum delay paths.

Zhang *et al.*

The approach proposed by Zhang et al. [38] combines genetic algorithms and simulated annealing. Simulated annealing is used here to avoid premature convergence of the genetic algorithm.

The precedure of codingis the same as used in[8], but with a new initialization process of the population, which simulates the grafting process. This process is done in 2 steps:

• Trunk creation:

(a) The source vertex s joins the multicast tree and becomes the current vertex.

(b) Select a random edge e connected to the current vertex.

(c) The other endpoint of edge e joins the multicast tree and becomes the current vertex.

(d) Repeatsteps (2) - (3) until the current vertex becomes the destination vertex.

• Membrane connection

(a) Choose a random destination as the current vertex that is not yet included in the multicast tree.

(b) Select a random edge e connected to the current vertex.

(c) The other endpoint of the edge e joins the multicast tree and becomes the current vertex.

(d) If the current vertex is not included in the tree, repeat (2) - (3), otherwise go to (5)

(e) Repeat steps (1) - (4) until all destinations are included in the multicast tree.

• Selection: in this approach, a roulette selection method is adopted. The last chromosome of the population is used to save the best individuals of all generations, in order to ensure the convergence of the algorithm towards a global optimal solution.

• Crossover: the crossover steps are as follow:

(a) Save the common edges of parent trees. Several separate subtrees and isolated nodes are then obtained.

(b) Randomly add edges of the non-connected destinations to the source and then connect separate subtrees and nodes.

(c) Removeprotrudingnodes and edges.

- Simulated annealing mutation: the mutation operation proposed in [38] is not the one commonly used in genetic algorithms as it uses simulated annealing.

An approximate solution is accepted with a probability p_m:

$$p_m = \begin{cases} \exp\frac{-(\text{fit}-\text{fit}')}{t} & \text{fit} > fit' \\ 1 \quad \text{fit} \leq \text{fit}' \end{cases} \quad (7)$$

where:

- fit: the fitness value of parent
- fit′: the fitness value of the son

If $\text{fit} \leq \text{fit}'$, i.e. the mutation-created son is better than the parent, it must be copied directly into the next generation. Otherwise, the offspring is accepted by the probability p_m.

The value of the probability p_m depends on the temperature t. This temperature should be initialized by a relatively large value so that the approximate solutions are accepted with a high probability, and this will ensure the convergence of the algorithm.

b) Taboo search

Taboo search is a metaheuristic formulated for the first time by Glover in 1986 [39], which guide other heuristics in such a way that they explore various areas of the solution space. Thanks to this method we find a local minimum.

At each iteration of the taboo search method, we start with the current solution. We explore its "neighborhood", and from this neighborhood, we select the best possible

solution that does not necessarily improve the current solution. This new solution becomes the current solution, which is referred to as a "movement". The "neighborhood" of the solution is defined as a part of the solution space that is accessible by applying an elementary transformation. These transformations can be defined in many ways depending on the specifics of the problem.

Some problems arise when looking for the optimal solution in this way. Supposing that the current solution is a local optimum, the new solution thus obtained by the next move should be a little worse. The problem arises in the subsequent movement because we will fall back on the local optimum that we have just avoided. In next iterations, the search will alternate between these two solutions. This is why the heuristic must have memory: the mechanism is to prohibit (hence the name of taboo) to return to the last explored positions. The taboo search method "memorizes" the elementary transformations applied to a number of the previous movements and places them in a "taboo list". Movements that are the opposite of those already executed are prohibited. This prevents to return to a solution already used and to find oneself in a local optimum.

TS-CST algorithm

The $TS-CST$ algorithm is a taboo search heuristic proposed by Skorin-Kapov and Kos [40], for the delay-constrained multicast routing problem.

The authors propose to reduce the size of the graph before the implementation of the algorithm by using some standard rules, with a slight modification due to the additional delay constraint. First, we prune the graph of all the vertices that are not destinations (vertices of the set $V\backslash D$) which are of degree 1, because they will surely not be included in the solution. On the other hand, it is observed that the edge adjacent to each vertex of destination which is of degree 1, will always be included in the multicast tree.

Potential solutions are multicast trees represented by strings of $|V\backslash D|$ bits, where each bit corresponds to a vertex in $V\backslash D$. If a bit is equal to 1, it means that the corresponding vertex is not included in the tree. For vertices belonging to the tree, the corresponding bits are equal to 0.

Potential solutions are multicast trees represented by $|V\backslash D|$ bits, where each bit corresponds to a vertex in $V\backslash D$. If a bit equals 1, it means that the corresponding vertex is not included in the tree. For the vertices belonging to the tree, the corresponding bits are equal to 0.

The evaluation of a potential solution starts by eliminating the vertices corresponding to the bits set to 1, starting from graph G as well as all the adjacent edges. The second step of the evaluation is to find a spanning tree with minimum cost, under delay constraint, of this modified graph. In this approach, the authors propose a modification of the Prim algorithm [33], to find a minimal cost spanning tree, where the delay between the source and each destination is less than Δ. Prim's algorithm consists of growing the tree from the source, and then at each step adds a minimum weight edge adjacent to the tree under construction. The procedure ends when all vertices are included in the tree. To ensure that the delay constraint is respected, the construction of the tree begins with the source vertex. The procedure ends when all vertices are included in the tree. At each iteration of the $TS-CST$ algorithm, to explore a neighborhood, we calculate the cost of each minimal spanning tree, under delay constraint, found for each neighboring solution.

As mentioned previously, the solutions that are similar to the current one are all that can be achieved by applying a basic transformation to the current solution. In this heuristic, we use a set of $|V\backslash D|$ elementary transformations, where the n^{th} transformation is defined as the modification of the value of the rank bit n, in the string that represents the current solution. According to this definition, the neighborhood of the current solution is the set of all bit strings that differ from the current one-bit solution chain. This basically means that in a "move", the new solution can only add or remove a multicast vertex from the previous solution. This procedure can be formulated as follows: if $X_{i-1} = x_1^{(i-1)} x_2^{(i-1)} \dots x_{|V\backslash D|}^{(i-1)}$ is a string of bits, where each x_k, $k = 1,2,\dots |V\backslash D|$ represents the k^{th} vertex in all $V\backslash D$. If we apply an elementary transformation m_n, $n = 1,2,\dots |V\backslash D|$, to X_{i-1}, we get:

$$X_i = x_1^{(i)} x_2^{(i)} \dots x_{|V\backslash D|}^{(i)} = m_n(X_{i-1}) = x_1^{(i-1)} x_2^{(i-1)} \dots x_{n-1}^{(i-1)} \overline{x_n^{(i-1)}} x_{n+1}^{(i-1)} \dots x_{|V\backslash D|}^{(i-1)} \quad (8)$$

In i^{th} "movement", we choose the solution X_i which has the minimal cost spanning tree, obtained by applying an elementary transformation on X_{i-1}, which is not in the taboo list. This solution is compared to the best solution found, and the best of both is maintained. The taboo list is updated after each move by adding the inverse of the elemental transformation applied to X_{i-1} to get X_i and eliminating the oldest member of the list.

The initial configuration of the $TS-CST$ algorithm is such that all bits are set to zero. This means that the multicast tree that matches the initial configuration includes all the vertices of the set $V\backslash D$. In other words, all the vertices of the set $V\backslash D$ are multicast vertices.

The choice of the number of iterations depends on the decision maker. Of course, a large number of iterations improves the quality of the solution obtained, but the execution time increases continuously with the number of vertices in the graph and with the value of the delay constraint.

TSDLMRA algorithm

In the TSDLMRA algorithm, proposed by Wang et al. [12], a solution is encoded by a vector x, of $|M|$ elements, where each element represents a path from the source s to a destination $u \in D$ of multicast tree, i.e $x = (p_1, p_2, \dots, p_k)$ with $k = |M|$, $p_i = p(s, u_i)$, $u_i \in U$, $1 \le i \le k$. The initial solution T_0 represents the tree of

the shortest paths. The best solution T_{best} is stored throughout the procedure, if T_0 satisfies the delay constraint, the best solution T_{best} is the initial solution T_0, otherwise, T_{best} is the minimal delay tree built by the Dijkstra algorithm [41].

c) Ant colony algorithm

The ant colony algorithms [42], [43] mimic the behavior of ant colonies in the real world. Without vision, ants can find the shortest path from the food source to their nests. During this time, they can be adapted to changing environment. Indeed, information is transmitted between ants through a substance called "pheromone". In the process of movement, an ant can not only leave the substance on the path over which it has passed, but also detect the existence and intensity of it, and move towards the direction where the intensity is high. As a result, the collective behavior of the ant colony indicates positive feedback from the information: more ants pass on a certain path, more likely that ants select this path to return later. Through the communication of information between ants, they make the optimal choice, thus reaching the goal of foraging.

Concerning the application of ant colony algorithms to the constrained multicast routing problem, several methods are proposed in [44], [45], [46], [47], and [48]. The common feature of these algorithms is that we first look for the shortest paths that connect the source vertex to each destination vertex, these paths are then merged to build the multicast tree, and the pheromone is updated according to the optimal path. This process is repeated until the algorithm converges. Despite the advantages of being parallel and distributed, this embodiment is inefficient due to the repetitive operation.

TGBACA algorithm

To improve the efficiency of ant colony algorithms, Wang *et al*. [49] propose a construction of the TGBACA ant colony algorithm. This algorithm optimizes directly the multicast tree.

This method search a minimal cost multicast tree under 4 constraints: bandwidth, delay, packet loss rate, and delay jitter.

In this approach, the communication network is modeled by a directed graph. the existence of an edge $e = (u, v)$ implies the existence of the edge $e' = (v, u)$. Generally, $C(e) \neq C(e')$, $D(e) \neq D(e')$, $L(e) \neq L(e')$, and $B(e) \neq B(e')$, where C, D, L, and B represent the cost, delay, packet loss rate, and bandwidth associated to each edge, respectively.

In this algorithm the ant does not select a single destination, but it aims to find a tree containing all the vertices of destination. Each vertex in the subtree that each ant sought is susceptible to be the current vertex.

The multicast tree $T = (E_T, V_T)$ is initialized by $E_T = NULL$ and $V_T = \{s\}$ where s is the source vertex. We create a set of edges E', initialized by $E' = \{e(s, i)\}$, where $e(s, i)$ is an edge belonging to the initial topology, and satisfying the following inequalities: $B(e(s, i)) \geq B_d$, $D(e(s, i)) \leq \Delta_d$, $1 - (1 - L(e(s, i))) \leq L_d$ where B_d, Δ_d and L_d represent respectively the bandwidth constraint, the delay constraint and the packet loss rate constraint.

The probability of selecting an edge of the set E' is given by the following formula:

$$P_i = \begin{cases} \frac{[\tau_i]^{\alpha}[\lambda_i]^{\beta}}{\sum_{e_j \in E'}[\tau_i]^{\alpha}[\lambda_i]^{\beta}} & \text{if} \quad e_i \in E' \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where e_i is the i^{th} edge of E, τ_i is the intensity of the pheromone of e_i, and λ_i is a function associated to e_i. λ_i can be a function of cost, delay, or packet loss rate depending on the case. We choose $\lambda \frac{1}{\cos t_i}$, where $\cos t_i$ is the cost associated to e_i, α and β are parameters used to adjust the effect of the pheromone intensity as well as the function λ_i.

Assuming that the edge $e(i, j)$ has been chosen, the following steps are followed:

- **Step 1**: add the edge $e(i, j)$ to the set of edges E_T of T and add the vertex j to the set of vertices V_T of T, ie $E_T = E_T \cup \{e(i, j)\}$, $V_T = V_T \cup \{j\}$;
- **Step 2**: modify the value of the delay $D(P_T(s, j))$ of the path connecting the source vertex and the vertex j, ie $D(P_T(s, j)) = D(P_T(s, i)) + D(e(i, j))$;
- **Step 3**: change the value of the packet loss rate $L(P_T(s, j))$ of the path connecting the source vertex and the vertex j, ie $L(P_T(s, j)) = 1 - (1 - L(P_T(s, i)))(1 - L(e(i, j)))$;
- **Step 4**: modify the value of the bandwidth $B(P_T(s, j))$ of the path connecting the source vertex and the vertex j, ie $B(P_T(s, j)) = \min(B(P_T(s, i)), B(e(i, j)))$;
- **Step 5**: update the set $E' = E' - E_1 + E_2$, where $E_1 = \{e(k, j) \backslash e(k, j) \in E'\}$ denotes the set of edges that have the vertex j as endpoint and belong to E', and the set $E_2 = \{e(j, k) \backslash k \notin V_T$, and $B(e(j, k) \geq B_d)$, $D(P_T(s, j)) + D(e(j, k)) \leq \Delta_d$ and $1 - (1 - L(P_T(s, j))(1 - L(e(j, k))) \leq L_d\}$ designates the edges that have j as endpoint and satisfy the quality of service constraints.

The deletion of the edges belonging to E_1 from the set E' guarantees that the degree of the added vertices is equal to 1 and avoids the appearance of loops. E_2 guaranteed that adding edges will satisfy quality of service constraints.

The authors propose the pheromone update formula as follows:

$$\tau_{ij} = (1 - \rho).\tau_{ij} + \rho.\Delta\tau_{ij} \tag{10}$$

where ρ represents an evaporation parameter used to control the rate of pheromone evaporation, and $\rho \in [0,1]$; $\Delta\tau_{ij}$ represents the pheromone increase on the edge $e(i, j)$.

The complexity of this algorithm is $\Theta(iter \times (antnum \times (|E| \times |V| + |E|) + |E|))$ where antnum is the number of ants, iter the number of iterations, $|E|$the number of edges, and $|V|$ the number of vertices in the initial graph.

NACOg algorithm

The NACOg algorithm, proposed by Yin et al.[15], looks for a minimal cost multicast tree, under delay and bandwidth constraints, in a directed graph. The authors first present a CTT strategy (Constrained tree traversal strategy), which ensures that the found trees always respect constraints. This strategy puts in place appropriate mechanisms to avoid the production of a loop, which will prevent the construction of a feasible tree. There are two cases of loop generation: the auto-cross loop, which occurs when the current path reaches a vertex that was previously visited by the same routing procedure, and the inter-cross loop that occurs when the path current reaches a vertex that is contained in a previously built path. The NACOg algorithm works with the CTT strategy to generate a minimal cost multicast tree, and drive scalable computation to minimize the cost of transmission. Each ant in a colony adopts this strategy to ensure the feasibility of the resulting multicast tree. The quality of this tree in terms of transmission cost is improved through the evolutionary optimization process by the use of two factors: the τ pheromone matrix and the η visibility matrix.

E. Artificial Neural Networks for the Multicast Routing Problem

Neural networks are also proposed to solve the problem of multicast routing, because of the important property of parallelism that they offer. Rauch and Winarske [19] proposed a modification of the neural networks for the travelling salesman problem to solve the routing problem. In 1982, Hopfield introduced Hopfield's neural network [20], since then many researchers have explored $HNNs$ (Hopfield Neural Networks): Pornavalai et al. [21] proposed modifying the $HNNs$ to solve this constrained problem, but Gee and Prager [22] showed that they are not efficient for large networks. Nozawa [28] and Jain and Sharma [37] proposed a $HNNs$ modification by adding other properties. Wang et al. [48] proposed transiently chaotic neural networks for this problem.

a) Rauch and winarske

Rauch and Winarske [19] propose a modification of the neural networks for the travelling salesman problem to solve the problem of routing. The authors took the case of a network that has nodes modeled by a square matrix C of order 16 ,where the coefficients c_{ij} represent the capacity (cost) between the nodes i and j. In this representation, it is assumed that each link has a capacity of 8 units, with maximum capacity. If $c_{ij} = 0$, it means that there is no direct link between i and j. The topology of this network requires that communication between 2 nodes requires at most 4.

Suppose we want to find a multicast path between node 1 and node 13 that go accross at most by 4 edges. A candidate path is the path that goes accross the $1-4-9-16-13$ nodes. In order to represent the 5 nodes, we use 5 vectors of dimensions 16: U_1, U_2, U_3, U_4 and U_5, where U_1 has zeros everywhere except the element 1, U_2 has zeros everywhere except element 4, and so on up to the vector U_5 which has zeros everywhere except element 13. The neural network algorithm starts with a reasonable initialization of the five U control vectors, then tries to converge to the final value of the control vectors, which corresponds to the five desired nodes.

b) Pornavalai et al.

Pornavalai *et al*. [21] proposed a modification of the Hopfiled neural networks to solve the delay constrained routing problem. The authors assume that a communication network is represented by a directed graph, where there is at most one arc connecting 2 vertices. We associate 2 parameters to each arc: cost and delay. These parameters are previously known by the source before searching the multicast path.

The Hopfield neural network is a discrete time recurrent neural network model, whose matrix of connections is symmetrical and has zeros on the diagonal, and where a single neuron is updated at each unit of time.

The most used activation function between the input U_i and the output V_i of each neuron in the network is the Sigmoid function:

$$V_i = g(U_i) = \frac{1}{1+e^{-\lambda_i \cdot U_i}} \quad (11)$$

The neuron outputs will be transmitted to the input of other neurons through the matrix of connections $T = [T_{ij}]$. The U_i input of a neuron is the weighted sum of all the data provided by the other neurons. The external input I_i represents the data provided by the user to the network. The dynamics of i^{th} neuron is defined by the following function:

$$\frac{dU_i}{dt} = -\frac{U_i}{\tau} + \sum_{j=1}^{N} T_{ij} V_j + I_i \quad (12)$$

If the gain of the amplifiers is relatively large, then (12) follows a gradient descent of the quadratic energy function:

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} T_{ij} V_i V_j - \sum_{i=1}^{n} I_i V_i \quad (13)$$

From the equations (11) and (12), the dynamics of the i^{th} neurone becomes:

$$\frac{dU_i}{dt} = -\frac{U_i}{\tau} - \frac{\partial E}{\partial V_i} \quad (14)$$

In this article, the authorsmodifiy in two steps the neural network proposed by Ali and Kamoun [50], which solves the unconstrained unicast routing problem. The first step is to add the delay constraint, and the second step is to turn the unicast problem into a multicast problem.

In this article, the authors modifiy in two steps the neural network proposed by Ali and Kamoun [50], which solves the unconstrained unicast routing problem. The first step is to add the delay constraint, and the second step is to turn the unicast problem into a multicast problem.

The first step is done by adding the delay term to the energy function of the unconstrained model. In the second step, the energy function thus obtained is modified so that the Hopfield network minimizes the cost of the multicast tree and not unicast paths independently. We use n matrices of dimensions $n \times n$, where n is the size of the network. The q^{th} matrix isused to find the unicast path to the destination q under time constraint. Each neuron in the matrix is described by a double index: (x, i), where x and i represent the row and the column respectively. The neuron in the line x and the column i represents the link between the nodes x and i. The cost term of the energy function corresponding to each destination m of the set D is defined as follows: the cost term of the energy function corresponding to each destination m of the set D is defined as follows:

$$E_1^m = \sum_{x=1}^{n} \sum_{i=1 i \neq x}^{n} C_{xi}.f_{xi}^m(V).V_{xi}^m \quad (15)$$

$$f_{xi}^m = \frac{1}{1+\sum_{j=1 j\neq x, j\in D}^{n} V_{xi}^j} \quad (16)$$

where $V = \{V_{xi}^k : k = 1,2,\ldots,n, k \neq m, k \in D\}$ and $V_{xi}^m = 1$ if the link between x and i is included in the multicast tree of destination m and $V_{xi}^m = 0$ otherwise.

$C_{xi} =$ is the cost of the link between x and i

The term of energy function that forces the output of the neuron to 1 or 0 is defined as follows:

$$E_2^m = \sum_{x=1}^{n} \sum_{i=1 i\neq x}^{n} V_{xi}^m(1 - V_{xi}^m) \quad (17)$$

The term of energy function that ensures that each matrix finds a full and continuous path from the source to a destination is defined as follows:

$$E_3^m = (1 - V_{ms}^m) + \sum_{x=1}^{n} \{\sum_{i=1 i\neq x}^{n} V_{xi}^m - \sum_{i=1 i\neq x}^{n} V_{ix}^m\}^2 \quad (17)$$

The term of energy function that penalizes neurons respresenting non-existing links in the network is defined as follows:

$$E_4^m = \sum_{x=1}^{n} \sum_{i=1, i\neq x, (x,i)\neq(m,s)}^{n} P_{xi}V_{xi}^m \quad (18)$$

where $P_{xi} = 1$ if the link between x and i is non − existing in the network and $P_{xi} = 0$ otherwise.

The delay constraint is expressed by the following inequality:

$$\sum_{x=1}^{n} \sum_{i=1, i\neq x, (x,i)\neq(m,s)}^{n} L_{xi}V_{xi}^m \leq \Delta \quad (19)$$

where:

$L_{xi} =$ the delay of link between x and i

To transform the inequality of delay constraint into equality, we use a neuron that has as function of transfer:

$$h(z) = \begin{cases} 0 & if: z \leq 0 \\ z & otherwise \end{cases} \quad (20)$$

where

$$z = \sum_{x=1}^{n} \sum_{i=1 i\neq x(x,i)\neq(m,s)}^{n} L_{xi}V_{xi}^m - \Delta \quad (21)$$

This neuron contributes positively only if the delay constraint is violated.

The term of energy function relative to the delay constraint can be defined as follows:

$$E_{5,LP}^m = H(z) \quad (22)$$

where:

$$H(z) = \int h(z)\mathrm{d}z \quad (23)$$

The total energy function for the delay-constrained multicast routing problem is defined as follows:

$$z = \sum_{m=1 m\in D}^{n} E_m \quad (24)$$

where E_m is the energy function of matrix m, used to find the unicast path connecting source s to destination m and is defined by:

$$E^m = E_1^m + E_2^m + E_3^m + E_4^m + E_{5,LP}^m \quad (25)$$

$$\begin{aligned} E^m = {} & \frac{\mu_1}{2}\sum_{x=1}^{n} \sum_{i=1 i\neq x(x,i)\neq(m,s)}^{n} C_{xi}.f_{xi}^m(V).V_{xi}^m \\ & + \frac{\mu_2}{2}\sum_{x=1}^{n} \sum_{i=1 i\neq x}^{n} V_{xi}^m(1 - V_{xi}^m) \\ & + \frac{\mu_3}{2}(1 - V_{ms}^m) \\ & + \frac{\mu_4}{2}\sum_{x=1}^{n} \{\sum_{i=1 i\neq x}^{n} V_{xi}^m - \sum_{i=1 i\neq x}^{n} V_{ix}^m\}^2 \\ & + \frac{\mu_5}{2}\sum_{x=1}^{n} \sum_{i=1, i\neq x, (x,i)\neq(m,s)}^{n} P_{xi}V_{xi}^m + \frac{\mu_6}{2}H(z) \end{aligned} \quad (26)$$

where $\mu_1, \mu_2, \mu_3, \mu_4, \mu_5,$ and μ_6 are coefficients used to specify and control the meaning of each term in the energy function.

From the equation (13), the evolution of the input voltage of U_{xi}^m is given by:

$$\frac{dU_{xi}^m}{dt} = -\frac{U_{xi}^m}{\tau} - \frac{\partial E^m}{\partial V_{xi}^m} \quad (27)$$

Replacing (26) in (27) gives the dynamic of the neural network:

$$\begin{aligned} \frac{dU_{xi}^m}{dt} = {} & -\frac{U_{xi}^m}{\tau} - \frac{\mu_1}{2}C_{xi}.f_{xi}^m(V).(1 - \delta_{xm}\delta_{is}) \\ & - \frac{\mu_2}{2}(1 - 2V_{xi}^m) + \frac{\mu_3}{2}\delta_{xm}\delta_{is} \\ & - \mu_4\sum_{y=1 y\neq x}^{n}(V_{xy}^m - V_{yx}^m) + \mu 4\sum_{y=1 y\neq i}^{n}(V_{iy}^m - V_{yi}^m) \\ & \frac{\mu_5}{2}P_{xi}(1 - \delta_{xm}\delta_{is}) \\ & + \frac{\mu_6}{2}L_{xi}(1 - \delta_{xm}\delta_{is})h(z) \end{aligned} \quad (28)$$

where:

$$V_{xi}^m = g_{xi}^m(U_{xi}^m =) = \frac{1}{1+e^{-\lambda_{xi}^m.U_{xi}^m}} \quad (29)$$

$$\delta_{ab} = \begin{cases} 1 & if: a = b \\ 0 & otherwise \end{cases} \quad (30)$$

The simulations have shown that the proposed neural networks, compared with other methods, generate solutions in the the neighborhood of optimal solutions, but Gee and Prager [22] have shown that they are not efficient for large networks. .

F. Optimization of the Multicast Routing Problem

As mentioned previously, the multicast routing problem is an optimization problem. In most of the methods proposed in the literature, this problem is formulated as a single objective optimization problem

(SOP), where a single objective function is considered ([5] ,[6] ,[7], [37] and [23]). recently, this problem has been dealt with in a multi-objective context (MOP), where several objective functions can be optimized (minimized or maximized) simultaneously ([24]-[27]). Table I compares several approaches of resolution, objective functions considered and constraint.

TABLE I: COMPARATIVE TABLE OF SEVERAL APPROACHES TO SOLVING THE MULTICAST ROUTING PROBLEM

Reference	Year	Objective Functions						Constraints				Optimization Problem	Approach of Resolution
		Cost	Delay	Bandwidth	Packet Loss Rate	Link Status	Power	Delay	Bandwidth	Delay jitter	Packet Loss Rate		
[3]	1991	✓										SOP	Exact Algorithm
[32]	1981	✓										SOP	Heuristic
[5]	1993	✓						✓				SOP	Heuristic
[7]	1998	✓						✓				SOP	Heuristic
[51]	2000	✓						✓				SOP	Genetic Algorithms
[8]	2001	✓						✓	✓			SOP	Genetic Algorithms
[38]	2009	✓						✓	✓	✓		SOP	Genetic Algorithms and Simulated Annealing
[40]	2002	✓						✓				SOP	Taboo Search
[12]	2004	✓						✓				SOP	Taboo Search
[44]	2011	✓						✓	✓	✓	✓	SOP	Ant Colonies Algorithm
[15]	2014	✓						✓	✓			SOP	Ant Colonies Algorithm
[21]	1995	✓						✓				SOP	Neural Networks
[52]	2014	✓						✓				SOP	Neural Networks in Fixed Time
[23]	2016	✓					✓	✓				SOP	Neural Networks in Fixed Time
[53]	2002		✓	✓								SOP	Genetic Algorithms
[54]	2003		✓	✓	✓			✓				SOP	Genetic Algorithms
[26]	2004	✓	✓		✓							SOP	Genetic Algorithms
[24]	2004	✓	✓			✓						SOP	Genetic Algorithms
[55]	2016	✓			✓			✓	✓			SOP	Neural Networks in Fixed Time

VI. CONCLUSIONS

The main objective of this article was to expose the various approaches to solving the problem of multicast routing in communication networks proposed in the literature. For this, we first saw the mathematical modeling of this problem, we then briefly described the theory of routing, introducing the 4 types of routing: unicast, anycast, broadcast and the type of routing traited in this document which is the multicast routing.

For this type of routing, different resolution approaches are adopted. The exact algorithms give an exact construction of the routing tree, but have many disadvantages, including high complexity in the worst case. To remedy this problem, several heuristics have been proposed. We have emphasized that a multicast routing problem is an optimization problem, where one (or more) objective function (s) must be optimized, under several constraints.

REFERENCES

[1] R. S. Chang and C. D. Wang, "Improved www multimedia transmission performance in http/tcp over atm networks," *IEEE Transactions on Multimedia*, vol. 1, no. 3, pp. 278–290, 1999.

[2] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, 1996.

[3] C. H. Chow, "On multicast path finding algorithms," in *Proc. Tenth Annual Joint Conference of the IEEE Computer and Communications Societies*, 1991, pp. 1274–1283.

[4] H. F. Salama, D. S. Reeves, and Y. Viniotis,"Evaluation of multicast routing algorithms for real-time communication on high-speed networks," *IEEE Journal*

on Selected Areas in Communications, vol. 15, no. 3, pp. 332–345, 1997.

[5] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos,"Multicast routing for multimedia communication," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 286–292, 1993.

[6] F. Bauer and A. Varma, "Distributed algorithms for multicast path setup in data networks," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 181–191, 1996.

[7] X. Jia,"A distributed algorithm of delay-bounded multicast routing for multimedia applications in wide area networks," *IEEE/ACM Transactions on Networking*, vol. 6, no. 6, pp. 828–837, 1998.

[8] W.Zhengying, S.Bingxin, and Z. Erdun,"Bandwidth-delay-constrained least-cost multicast routing based on heuristic genetic algorithm," *Computer Communications*, vol. 24, no. 7, pp. 685–692, 2001.

[9] S. Y. Tseng, Y. M. Huang, and C. C. Lin, "Genetic algorithm for delay-and degree-constrained multimedia broadcasting on overlay networks," *Computer Communications*, vol. 29, no. 17, pp. 3625–3632, 2006.

[10] T. Lu and J. Zhu, "A genetic algorithm for finding a path subject to two constraints," *Applied Soft Computing*, vol. 13, no. 2, pp. 891–898, 2013.

[11] H. Youssef, A. Al-Mulhem, S. M. Sait, and M. A. Tahir, "QOS-driven multicast tree generation using tabu search," *Computer Communications*, vol. 25, no. 11-12, pp. 1140–1149, 2002.

[12] H. Wang, J. Fang, H. Wang, and Y. M. Sun,"Tsdlmra: An efficient multicast routing algorithm based on tabu search,"*Journal of Network and Computer Applications*, vol. 27, no. 2, pp. 77–90, 2004.

[13] W. L. Yang,"A tabu-search based algorithm for the multicast-streams distribution problem,"*Computer Networks*, vol. 39, no. 6, pp. 729–747, 2002.

[14] S. Y. Tseng, C. C. Lin, and Y. M. Huang, "Ant colony-based algorithm for constructing broadcasting tree withdegree and delay constraints,"*Expert Systems with Applications*, vol. 35, no. 3, pp. 1473–1481, 2008.

[15] P. Y. Yin, R. I. Chang, C. C. Chao, and Y. T. Chu,"Niched ant colony optimization with colony guides for qos multicast routing," *Journal of Network and Computer Applications*, vol. 40, pp. 61–72, 2014.

[16] H. Wang, Z. Shi, and S. Li, "Multicast routing for delay variation bound using a modified ant colony algorithm," *Journal of Network and Computer Applications*, vol. 32, no. 1, pp. 258–272, 2009."

[17] J. Nie, J. Wen, J. Luo, X. He, and Z. Zhou, "An adaptive fuzzy logic based secure routing protocol in mobile ad hoc networks," *Fuzzy Sets and Systems*, vol. 157, no. 12, pp. 1704–1712, 2006.

[18] B. L. Su, M. S. Wang, and Y. M. Huang, "Fuzzy logic weighted multi-criteria of dynamic route lifetime for reliable multicast routing in ad hoc networks," *Expert Systems with Applications*, vol. 35, no. 1-2, pp. 476–484, 2008.

[19] H. E. Rauch and T. Winarske, "Neural networks for routing communication traffic," *IEEE Control Systems Magazine*, vol. 8, no. 2, pp. 26–31, 1988.

[20] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[21] C. Pornavalai, G. Chakraborty, and N. Shiratori, "A neural network approach to multicast routing in real-time communication networks," in *Proc. International Conference on Network Protocols*, 1995, pp. 332–339.

[22] A. H. Gee and R. W. Prager, "Limitations of neural networks for solving traveling salesman problems," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 280–282, 1995.

[23] N. Saber, M. Mestari, and A. A. El Mahjoub, "The multi-constrained least-cost multicast problem with neural networks in fixed time," *Applied Mathematical Sciences*, vol. 10, no. 19, pp. 931–945, 2016.

[24] J. Crichigno and B. Barán, "Multiobjective multicast routing algorithm," in *Proc. International Conference on Telecommunications*, 2004, pp. 1029–1034.

[25] J. Crichigno and B. Barán, "Multiobjective multicast routing algorithm for traffic engineering," in *Proc. 13th International Conference on Computer Communications and Networks*, 2004, pp. 301–306.

[26] J. Crichigno and B. Barán, "A multicast routing algorithm using multiobjective optimization," in *Proc. International Conference on Telecommunications*, 2004, pp. 1107–1113.

[27] A. Roy and S. K. Das, "Qm 2 rp: AQos-based mobile multicast routing protocol using multi-objective genetic algorithm," *Wireless Networks*, vol. 10, no. 3, pp. 271–286, 2004.

[28] S. E. Deering and D. R. Cheriton, "Multicast routing in datagram internetworks and extended lans," *ACM Transactions on Computer Systems*, vol. 8, no. 2, pp. 85–110, 1990.

[29] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, 1988.

[30] K. I. Park, "QoS in packet networks," *Springer Science & Business Media*, vol. 779, 2004.

[31] S. E. Dreyfus and R. A. Wagner, "The steiner problem in graphs," *Networks*, vol. 1, no. 3, pp. 195–207, 1971.

[32] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," *ActaInformatica*, vol. 15, no. 2, pp. 141–145, 1981.

[33] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Labs Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.

[34] H. Takahashi and A. Matsuyama, "An approximate solution for the steiner problem in graphs," *Math. Japonica*, vol. 24, no. 6, pp. 573–577, 1980.

[35] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, no. 1, pp. 48–50, 1956.

[36] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence*, Ann Arbor, MI: University of Michigan Press, 1975.

[37] S. Jain and J. D. Sharma, "Delay bound multicast routing using hopfield neural network," *International Journal of computer Theory and Engineering*, vol. 2, no. 3, p. 384, 2010.

[38] L. Zhang, L. Cai, M. Li, and F. Wang, "A method for least-cost QOS multicast routing based on genetic simulated annealing algorithm," *Computer Communications*, vol. 32, no. 1, pp. 105–110, 2009.

[39] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers &Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.

[40] N. Skorin-Kapov and M. Kos, "The application of steiner trees to delay constrained multicast routing: Atabu search approach," in *Proc. 7th International Conference on Telecommunications*, 2003, pp. 443–448.

[41] E. W. Dijkstra, "A note on two problems in connexion with graphs,"*Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
[42] A. Maniezzo, M. Colorni, and V. Dorigo, "Distributed optimization by ant colonies," in *Proc. Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. Mit Press, 1992, p. 134.
[43] M. Dorigo, "Optimization, learning and natural algorithms," PhD. thesis, Politecnico di Milano, Italy, 1992.
[44] H. Wang, H. Xu, S. Yi, and Z. Shi, "A tree-growth based ant colony algorithm for qos multicast routing problem," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11787–11795, 2011.
[45] B. Gong, L. Li, and X. Wang, "Multicast routing based on ant algorithm with multiple constraints," in *Proc. International Conference on Wireless Communications, Networking and Mobile Computing*, 2007, pp.1945–1948.
[46] K. Verbeeck and A. Nowe, "Colonies of learning automata," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 6, pp. 772–780, 2002.
[47] G. Lu and Z. Liu, "Multicast routing based on ant-algorithm with delay and delay variation constraints," in *Proc. IEEE Asia-Pacific Conference on Circuits and Systems*, 2000, pp. 243–246.
[48] J. Yang, X. Shi, M. Marchese, and Y. Liang, "An ant colony optimization method for generalized tsp problem," *Progress in Natural Science*, vol. 18, no. 11, pp. 1417–1422, 2008.
[49] H. Wang, H. Xu, S. Yi, and Z. Shi, "A tree-growth based ant colony algorithm for QoS multicast routing problem". Expert Systems with Applications, 38(9), 11787-11795, 2011.
[50] M. K. M. Ali and F. Kamoun, "Neural networks for shortest path computation and routing in computer networks," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 941–954, 1993.
[51] R. H. Hwang, W. Y. Do, and S. G. Yang, "Multicast routing based on genetic algorithms," *J. Inf. Sci. Eng.*, vol. 16, no. 6, pp. 885–901, 2000.
[52] N. Saber, M. Khouil, and M. Mestari, "Delay constrained multicast tree with neural networks in fixed time," *International Journal of Engineering Sciences &Research Technology*, 2014.
[53] A. Roy, N. Banerjee, and S. K. Das, "An efficient multi-objective qos-routing algorithm for wireless multicasting," in *Proc. IEEE 55th Vehicular Technology Conference*, 2002,pp. 1160–1164.
[54] X. Cui, C. Lin, and Y. Wei, "A multiobjective model for qos multicast routing based on genetic algorithm," in *Proc. International Conference on Computer Networks and Mobile Computing*, 2003, pp. 49–53.
[55] N. Saber and M. Mestari, "Multiobjective optimization for qos multicast routing problem with neural networks in fixed time," *International Journal of Computer Science and Information Security*, vol. 14, no. 8, pp. 186–197, 2016.

Nadia SABER received the engineering degree in mathematics and computer science applied to economy from Hassan II university, Faculty of science and techniques Mohammedia, in 2012 and the Ph.D. degree in mathematics and computer science from Hassan II university, ENSET Mohammedia, in 2017. She is currently an assistant Professor in EMSI Casablanca, and a member of the LPRI laboratory and SSDIA Laboratory. Her research interests include neural networks for multicast routing, neural networks hardware implementation, high-speed techniques and systems for neural networks, and solving optimization problems using neural networks.

Mohammed Mestari received the M.A. degree from ENSET, Mohammedia, Morocco, in 1991 and the Ph.D. degree in applied mathematics and the Ph.D. degree in artificial intelligence from Hessan II University, Faculty of Science Ben Mâ€™Sik Casablanca, in 1997 and 2000, respectively. He is currently a Professor of Applied Mathematics at ENSET, Mohammedia, and a head of the artificial intelligence research team affiliated with the laboratory of signals, distributed systems and artificial intelligence. His research interests include neural networks for signal processing, neural networks hardware implementation, high-speed and/or low-power techniques and systems for neural networks, and theoretical issues directly related to hardware implementation.