

OPTIMIZATION PROBLEMS IN MULTICAST TREE CONSTRUCTION

C.A.S. OLIVEIRA, P.M. PARDALOS, AND M.G.C. RESENDE

ABSTRACT. Multicasting is a technique for data routing in networks that allows multiple destinations to be addressed simultaneously. The implementation of multicasting requires, however, the solution of difficult combinatorial optimization problems. In this chapter, we discuss combinatorial issues occurring in the implementation of multicast routing, including multicast tree construction, minimization of the total message delay, center-based routing, and multicast message packing. Optimization methods for these problems are discussed and the corresponding literature reviewed. Mathematical programming as well as graph models for these problems are discussed.

1. INTRODUCTION

Routing is a fundamental task in network systems in general. Any computer network has as one of its main functions to send information (usually in the form of packets) between clients and servers. Techniques used for routing in traditional network systems can be classified in the following way:

- *Unicast routing*: under this routing paradigm, the objective is to transport data packets between single sources and destinations. This is the simplest routing method, and therefore is also the most frequently used. The idea of single source-destination transport of information has been explored in countless applications, and allowed the development of protocols such as `ftp`, `http`, which are used in the implementation of important Internet applications.
- *Broadcast routing*: in the broadcast paradigm of routing, a network node can send packets to all nodes directly connected to it. Broadcasting is popular in mass distribution media such as radio and TV, but it has also applications in computer networks. For example, broadcasting is generally used in packet switching networks for distributing data in the local segment, under network implementations such as Ethernet, which use a bus topology. In the case of Ethernet networks, client nodes are addressed as part of a network segment using broadcast messages (in an IP implementation, this happens when IP addresses ending in 255 are used).

Both paradigms for packet routing, although being useful, are not adequate when information needs to be sent to a relatively large group of users, that are geographically separated and have similar interest on a piece of data. This situation has led to the development of network protocols capable of sending information to selected destinations [Deering and Cheriton, 1990]. Such protocols are called *multicast routing* protocols. Different implementations have been proposed in the last 20 years, with varying levels of success.

Date: June 29, 2005.

Key words and phrases. Multicasting, data routing, networks, combinatorial optimization, multicast tree construction, delay minimization, cache placement, center-based multicast routing, and multicast packing.

AT&T Labs Research Technical Report TD-6DTLXR. To appear in *Handbook of Optimization in Telecommunications*, M.G.C. Resende and P.M. Pardalos (eds.), Springer, 2005.

However, all multicast routing implementations share similar requirements, in terms of the distributed combinatorial optimization problems that must be solved.

In this chapter, we discuss some of the most important combinatorial optimization problems occurring in multicast routing systems. We provide formulations for these problems and discuss some of the techniques that have been used to solve them in practice, as well as remaining open problems in this area.

1.1. Multicast network concepts. A *multicast group* is a set of network clients and servers interested in sharing a specific set of data. A multicast protocol has the objective of connecting members of the multicast group in an optimal way, by reducing the amount of bandwidth necessary but also considering other issues such as communication delays and reliability.

Multicast systems have in common the fact that the total amount of resources used by the network becomes large as the number of members in the multicast group increases. This makes it difficult to provide services at desirable levels, unless efficient algorithms are used to reduce resource consumption.

The delivery of data to a selected number of users organized as a multicast group is called *multicast routing*. This concept was introduced by Wall [1980] in order to allow the implementation of network services that require addressing specific sets of users. Despite having been proposed back in the 1980s, implementations of the multicast concept have not been brought into practical working until the the 1990s. Even so, multicast routing is not available by default in most routers, since it has the potential to slow down network servers. This shows the need for algorithms allowing efficient implementation of multicast routing concepts.

Multicast systems can be classified according to the types of groups involved. A multicast group can be *sparse* when the number of sparse group elements in the group is small compared to the total size of the network. If this is not the case, then the group is called a *pervasive group* [Waitzman et al., 1988]. Another classification of multicast groups is according to the duration of the required connection and the possibility of changes in the group membership. A *static group* is one that is fixed at its creation time, and cannot be changed afterwards. A *dynamic group*, on the other hand, is allowed to change by adding or removing nodes.

Several protocols have been proposed to implement the multicast network concept. Among the most important techniques we cite the following:

- DVMRP [Deering and Cheriton, 1990, Waitzman et al., 1988] (Distance-Vector Multicast Routing Protocol) is a protocol used to integrate multicast routing to RIP (interior gateway protocol). It allows tunneling of multicast traffic over standard routers.
- The MOSPF [Moy, 1994a] is an extension of the traditional OSPF [Moy, 1994b, Thomas II, 1998] (Open Shortest Path First) to handle multicast applications. Thus, it is able to integrate into existing routers.
- PIM [Deering et al., 1996] (protocol independent multicast) is a protocol that tries to explore sparsity in multicast groups. It is also designed to be independent of the underlying unicast routing protocol.
- Core-based trees [Ballardie et al., 1993] are a technique to allow easier creation of multicast trees, by the use of a core that can be reused by different multicast groups. It is a protocol that can be classified among other shared tree technologies [Chiang et al., 1998, Wei and Estrin, 1994].

- MBONE [Eriksson, 1994]: The Internet multicast backbone (MBONE) was introduced as a implementation of multicasting capable of sending live streams of video and sound using an existing Internet connection. In fact, one of the first applications of the MBONE was the transmission of IETF (Internet Engineering Task Force) meetings.

In unicast routing systems, routes are computed using classical algorithms such as Bellman [1957] and Dijkstra [1959]. Information about best routes are stored in tables in the routers' main memory. These algorithms have polynomial complexity, generally in the order of $O(n^3)$. Algorithms for routing in multicast networks, on the other hand, cannot provide exact solutions in polynomial time, since the underlying problems are NP-hard, as seen in the next sections. Some of these problems have been studied by many researchers in the last years, starting with Dalal and Metcalfe [1978].

Several documents discussing the design and implementation of current multicast protocols are available on the Internet. For example, the Internet task force draft [Semeria and Maufer, 1996] on multicast technologies is freely available. For papers and surveys on general aspects and algorithms related to multicast systems, see Du and Pardalos [1993a], Wan et al. [1998], Pardalos et al. [1993], Pardalos and Khoury [1995], Paul and Raghavan [2002], and Salama et al. [1997b].

In this chapter, we review algorithmic and modeling strategies used to solve problems in multicast routing. Among the problems discussed, are the multicast tree construction, delay minimization, cache placement problems, center-based multicast routing, and multicast packing.

1.2. Applications. Several applications can benefit of the use of multicast routing schemes. We list some possible scenarios, occurring in areas such as corporate environments, education, entertainment, and real-time communication.

A possible scenario of application of multicast in corporate environments is the implementation of financial data delivery. In this application, users are interested in reliable and up to date information about stocks, bonds, and several financial services provided by the stock market. The advantage of multicast routing in this scenario is the possibility of reaching users that are registered to receive the information, with the group membership status being updated automatically, independent of the data source. This type of application requires reliable message delivery mechanisms, which is one of the main areas of research in multicast systems [IBM, 2005].

A second application that has received a lot of attention is video-conferencing [Eriksson, 1994, Sabri and Prasada, 1985]. The implementation of video-conferencing requires the use of fast protocols that can deliver information to a large number of users quickly. Multicast is an important technology for video-conferencing due to several reasons, including the savings in network bandwidth, the possibility of dynamically managing the set of users receiving the information, and the use of fast mechanisms for sharing the costs of connection. In fact, one of the first practical uses of a multicast protocol was the Internet multicast of sessions of IETF meetings.

Software delivery [Han and Shahmehri, 2000] is another example of application that can benefit of the use of multicast protocols. In particular, distribution of software in a company requires a large amount of bandwidth, the deliveries are quite predictable, and there is a strong motivation for allowing the computational burden to be shared among the set of computers being updated.

Group-ware collaboration [Chockler et al., 1996, Ellis et al., 1991] requires that a set of users keep data (such as calendars, working files, sets of archives in revision systems) in a

large number of locations, where it can be accessed by workers involved in a collaborative task. Group-ware requires the use of multicast protocols since it is very frequent that files need to be synchronized in this type of applications. Each set of files can be viewed as participating in a multicast group, and data is updated among elements of the group whenever necessary.

Finally, several other applications could benefit from the use of multicast protocols, including the delivery of multimedia content on the Internet [Pasquale et al., 1998], real-time video streaming [Jia et al., 1997, Kompella et al., 1996], and even networked games [Park and Park, 1997].

1.3. Notation. We use standard notation for graphs. A network is modeled as a graph $G = (V, E)$, where the set of nodes V represent servers, clients, or intermediate nodes, and the set of edges E represent links between nodes in V . A *multicast group* M is a set of source nodes S (also called servers), which store data that must be transferred to clients, together with a set of destinations D . All problems in multicast routing require that elements of the multicast group $M = S \cup D$ be connected in some specified way. In particular, we consider more commonly that S is composed of a single source node s .

The following functions defined on graph elements will be used throughout the paper. In particular, let $d(v)$, for $v \in V(G)$ represent the degree of a node, i.e., the number of nodes adjacent to v . Similarly, let $\delta(v)$ and $\Delta(v)$ be the minimum and maximum value of $d(v)$, for $v \in V$, respectively. A path between two nodes $a, b \in V$ is denoted by \mathcal{P}_{ab} . The length of \mathcal{P}_{ab} is the number of edges in the path.

1.4. Chapter organization. This chapter is organized as follows. In Section 2 we describe the multicast tree routing problem (MRP) which has the objective of finding an optimal tree linking the elements of the multicast group. This is a fundamental problem in multicast routing, and several models and algorithms have been proposed for its solution. In Section 3 we discuss the approaches to solve the MRP based on Steiner tree algorithms; we will see different ways of modeling the problem, including the use of center nodes and rings to improve the resulting solution. The need for network designs that support multicast in an efficient way lead to the formulation of the multicast packing problem, which will be discussed on Section 4. In Section 5, other problems related to multicast routing will be discussed, such as the point-to-point connection problem, and the cache placement problem. Finally, in Section 6 we provide some concluding remarks.

2. MINIMUM MULTICAST TREE ROUTING PROBLEM

2.1. Problem definition. One of the most difficult problems in multicast networks consists of finding routing paths linking the source (or set of sources) to the set of destinations, while simultaneously minimizing some cost function. A way of achieving this objective is computing a tree connecting the source and destination nodes. Thus, we are frequently interested in finding trees connecting sources and destinations with low total cost. The resulting problem is called the *multicast routing tree problem* (MRP).

A mathematical formulation of the MRP based on graphs can be given as follows. Let $G = (V, E)$ be a graph representing the network, $s \in V$ the source node, and $D \subset V$ the set of destinations. We assume that each node $v \in D$ must receive a copy of the information stored in s , and therefore $D \cup s$ form a multicast group. The set E of edges has an associated function $w : E \rightarrow \mathbf{R}_+$ returning the cost of using edge e . Then, the objective of the MRP is to find a tree $T \subseteq E$ connecting the source node s to each of the destinations $d \in D$, such that the total cost $\sum_{e \in T} w(e)$ is minimum.

The multicast tree problem has been studied by many researchers [Aguilar et al., 1986, Berry, 1990, Bauer, 1996, Ballardie et al., 1993, Baldi et al., 1997, Du and Pardalos, 1993a, Pardalos and Du, 1998, Frank et al., 1985, Feng and Yum, 1999, Im et al., 1997, Jiang, 1992, Wan et al., 1998, Pardalos et al., 2000] using strategies that range from simple heuristics to approximation algorithms. However, no exact polynomial algorithm is known for the problem in general, since it is NP-hard, as shown in the next theorem.

Theorem 1. *The multicast routing problem is NP-hard.*

Proof. By reduction from the Steiner problem on graphs. In the Steiner problem, we are given a graph $G = (V, E)$, a cost function $w : E \rightarrow \mathbf{R}_+$, a set of required nodes R , and a set of Steiner nodes $V \setminus R$. The objective is to find a tree T connecting all nodes in R at minimum cost.

The reduction is simple: take one of the nodes $v \in R$ to become the source, and make $D = R \setminus \{v\}$. Clearly, an optimal solution to this instance of the MRP instance will be also optimal for the Steiner problem. An optimal solution to the MRP can also be found given the optimum solution for the Steiner tree instance, thus the multicast routing problem in NP-hard. \square

When applying the MRP to specific problems, a number of additional constraints arise. The two most important additional constraints for the MRP are the following.

- *Delay constraints.* In applications that require real time coordination, such as video-conferencing and other multimedia systems, it is very important that the total delay in the routing of packets be as small as possible. In practice, it is required that the delivery of data take less than some delay threshold.
- *Reliability constraints.* In some multicast systems, an important concern is that data be delivered with some guarantee of reliability. This is done usually by requiring that additional capacity be reserved on each arc of the network, so that in the case of failure in some edge the data can be diverted through other paths in the network.

2.2. Flooding and reverse path-forwarding algorithms. The simplest way of sending packets to multiple nodes in a network is using the technique called *flooding*, which consists of recursively sending data to all neighbors of a node until the destinations are reached. The algorithm is shown in Algorithm 1. Note that this algorithm requires that each node keep track of packages that were sent through it during the process, otherwise a packet can loop indefinitely through the network. Keeping this extra information limits the usefulness of flooding, since the number of all packages that can be reliably sent is bounded by the available memory on any routing machine. There are also inevitable bandwidth losses in this process since many non-required edges are saturated with data. Finally, there is no confirmation that a package reached the required destinations.

To improve the performance of the flooding algorithm, the *reverse path-forwarding algorithm* (RPFA) was proposed by Dalal and Metcalfe [1978]. In the RPFA, each node is responsible for finding an edge that is in the path to the destination. In this way, traffic is not accepted from all nodes, but just from a selected number of neighbors. The edge from which traffic is accepted, when routing to a particular node, is called the *parent link*. A simple algorithm to determine the parent link for a node s is the following: let e be the first edge from which a package having origin s was received. From now on, if a package is received from an edge that is not considered to be the parent link for the given source node, then drop the package. Otherwise, receive the package and broadcast it to all

```

procedure Flooding-Algorithm
begin
  Receive packet  $p$  from node  $u$ ;
  if destination( $p$ ) =  $v$  then
    | PacketReceived;
  else
    | if packet was not previously processed then
    | | Send packet  $p$  to all nodes in  $N(v) \setminus \{u\}$ ;
  end
end
end

```

Algorithm 1: Flooding algorithm for node v

other neighbor nodes. Improvements to this basic algorithm are discussed by Semeria and Maufer [1996].

2.3. An integer programming formulation. In this section, we discuss an integer programming model for a generalized multicast routing problem, originally proposed by Ronha and Tobagi [1994]. The model is for a version of the MRP in which there are costs and delays for each link, and a set $\{1, \dots, T\}$ of multicast groups. Each group i has its own source s_i , a set of n_i destinations $d_{i1} \dots, d_{in_i}$, a maximum delay Δ_i , and a bandwidth request r_i . There is also a matrix $B^i \in R^{n \times n_i}$ for each group $i \in \{1, \dots, T\}$ of source-destination requirements. The value of B^i_{jk} is 1 if $j = s$, -1 if $j = d_{jk}$, and zero otherwise. The node-edge incidence matrix is represented by the matrix $A \in Z^{n \times m}$.

The considered network has n nodes and m edges. The vectors $W \in R^m$, $D \in R^m$ and $C \in R^m$ give respectively the costs, delays, and capacities for each link in the network. The variables in the formulation are X^1, \dots, X^T (where each X^i is an $m \times n_i$ matrix), and Y^1, \dots, Y^T (where each Y^i is a vector of m elements), and $M \in R^T$. The variable $X^i_{jk} = 1$ if and only if link j is used by group i to reach destination d_{ik} . Similarly, variable $Y^i_j = 1$ if and only if link j is used by multicast group i . Also, variable M_i represents the delay incurred by multicast group i in the current solution.

In the following formulation, the objectives of minimizing total cost and maximum delay are considered. However, the constant values β_c and β_d represent the relative weight given to the minimization of the cost and to the minimization of the delays, respectively. Using the variables shown, the integer programming formulation is given by:

$$(1) \quad \min \quad \sum_{i=1}^T r_i \beta_c C Y^i + \beta_d M_i$$

subject to

$$(2) \quad AX^i = B^i \quad \text{for } i \in \{1, \dots, T\}$$

$$X_{jk}^i \leq Y_j^i \leq 1$$

$$(3) \quad \text{for } i \in \{1, \dots, T\}, j \in \{1, \dots, n\}, k \in \{1, \dots, n_i\}$$

$$(4) \quad M_i \geq \sum_{j=1}^k D_j X_{jk}^i \quad \text{for } i \in \{1, \dots, T\}, k \in \{1, \dots, n_i\}$$

$$(5) \quad M_i \leq L_i \quad \text{for } i \in \{1, \dots, T\}$$

$$(6) \quad \sum_{i=1}^T r_i Y^i \leq C$$

$$(7) \quad X_{jk}^i, Y_j^i \in \{0, 1\}, \quad \text{for } 1 \leq i \leq T, 1 \leq j \leq K, 1 \leq k \leq n_i.$$

The constraints in the above integer program have the following meaning. Constraint (2) is the flow conservation constraint for each of the multicast groups. Constraint (3) determines that an edge must be selected when it is used by any multicast tree. Constraints (4) and (5) determine the value of the delay, since it must be greater than the sum of all delays in the current multicast group and less than the maximum acceptable delay L_i . Finally, constraint (6) says that each edge i can carry a value which is at most the capacity C_i . This is a very general formulation, and clearly cannot be solved exactly in polynomial time because of the integrality constraints (7).

This formulation was used by Ron Ha and Tobagi [1994] to derive exact algorithms for the general problem. Initially, a decomposition technique was used to divide the constraint matrix into smaller parts, where each part could be more easily solved. This part of the algorithm can be executed using some standard mathematical programming decomposition techniques, as shown e.g., in Bazaraa et al. [1990]. Then, a branch-and-bound algorithm was proposed to solve the resulting problem. The lower bounding procedure is important in this branch-and-bound implementation, since it can benefit from the efficiency gains of the decomposition method, resulting in improved computation time.

3. STEINER TREE PROBLEMS AND MULTICAST ROUTING

3.1. Steiner tree algorithms. Due to the similarity of the MRP and Steiner tree problems, one of the simplest and more effective techniques for solving the MRP involve careful modification of algorithms for Steiner tree [Bauer and Varma, 1997, Du et al., 2001, Du and Pardalos, 1993b, Hwang and Richards, 1992, Hwang et al., 1992, Kou et al., 1981, Takahashi and Matsuyama, 1980, Winter, 1987, Winter and Smith, 1992, Pardalos and Khoury, 1995, 1996, Pardalos et al., 1993], according to the necessary extra constraints. As an example of application of this solution scheme, we introduce the algorithm for the Steiner tree problem described in Algorithm 2, known as the KMB algorithm [Kou et al., 1981].

The KMB is an approximation algorithm which constructs a solution for the Steiner tree problem where the required nodes are elements of $D \cup s$ and the nodes in the required set are connected using the union of shortest paths. It is easy to prove the following result:

Theorem 2. *The KMB algorithm returns a solution within a factor 2 of the optimum.*

The KMB heuristic shown above provides a good scheme for heuristic development for the MRP, since it gives an initial guarantee of approximate optimality. Moreover, studies on the quality of solutions computed by the KMB heuristic [Wall, 1980] showed that the

```

procedure KMB-Algorithm
begin
  Let  $K = (V, E)$  be a complete graph with  $|D| + 1$  nodes;
  foreach  $(u, v) \in E(K)$  do
    |  $w(u, v) \leftarrow d(u, v)$ , where  $d$  is the minimum distance from  $u$  to  $v$  in  $G$ ;
  end
  Let  $T$  be a minimum spanning tree on  $K$ ;
  Let  $T'$  be the union of all shortest paths  $v \rightsquigarrow w$  in  $G$ , s.t.  $(v, w) \in T$ ;
  Let  $T''$  be a minimum spanning tree  $T'$ ;
  return  $T''$ 
end

```

Algorithm 2: The KMB algorithm for Steiner tree.

algorithm performs very well in problems occurring in practice, exceeding by a large difference the factor 2 guarantee described in Theorem 2. Similarly, Doar and Leslie [1993] have shown that this heuristic usually achieves at most 5% of difference to the optimal solution value, for many instances of large size.

The main challenge in using techniques based on KMB to solve the multicast routing problem consists of distributing the computational effort among all nodes involved in the multicast group. For this reason, many variants of the KMB algorithm have been proposed to solve the MRP, as described in the next section, to overcome the limitations of using a sequential, deterministic algorithm [Bharath-Kumar and Jaffe, 1983, Wall, 1982, Waxman, 1988, Wi and Choi, 1995].

An alternative algorithm to solve the Steiner tree problem was proposed by Takahashi and Matsuyama [1980], and is known as the *greedy incremental tree algorithm* (GIT). This heuristic is similar to the Prim's and Dijkstra's algorithm in the sense that it starts from a single node, and at each iteration it adds the edge that is closest to the current tree and which does not form a cycle. These steps are repeated until a tree connecting all required nodes is found. The GIT algorithm has also been adapted to the MRP and employed by some other researchers as an initial step in the construction of multicast trees [Carlberg and Crowcroft, 1997, Li and Mohapatra, 2003].

3.2. Steiner tree-based algorithms for the MRP with delay constraints. In this section, we describe some of the algorithms that have been proposed for the multicast routing problem [Im et al., 1997, Kompella et al., 1992, 1993b,a, Jia, 1998, Sriram et al., 1998]. Most of these algorithms share the following features:

- The Steiner tree model, as described in the previous section, is assumed;
- They are distributed versions of some of the well known heuristics for the Steiner tree problem;
- The distributed algorithms do not provide any guarantee of optimality;
- Unlike metaheuristic-based heuristics, these algorithms do not provide a guarantee of *local optimality*, which means that they do not check any neighborhood of the given solution for improvements that would locally improve the considered solution.

Im et al. [1997] proposed a distributed algorithm for the delay constrained multicast routing problem. In this distributed algorithm, the objective is to reduce the computational time for constructing the routing tree. The strategy used consists of creating the tree in only one round of data exchange, thus avoiding excessive message exchanges, which are the main time constraint on distributed algorithms. The basic steps in the algorithm can be

described as follows. Initially, messages are sent to nodes in the set $T = D \cup s$ composed of sources and destinations, with the objective of asking each node $v \in T$ to compute the node $w \in T \setminus v$ that is closest to v . Then, each node resends the messages to its children nodes, until the delay information has been computed by all nodes, and the results are sent back to the elements of T . When a node is selected to be part of the tree, it receives an ADD message and sends back an ADD_ACK message. The algorithm stops when all nodes in T have been added to the multicast tree. The following theorem has been proved in Im et al. [1997]:

Theorem 3. *The described algorithm produces a feasible multicast tree, and if there exists a feasible solution to the MRP, then the algorithm returns a feasible solution.*

Kompella et al. [1992] provided an algorithm based on a distributed version of the KMB heuristic. The basic modification included in the KMB algorithm is an added step, which guarantees that the constructed tree is feasible for the delay constraints. This works as follows: initially, a closure graph K is created from the original graph G , as in the original KMB algorithm. Then, at each step an edge of K is added to the tree. The edges are added, however, according to a function $g(e)$, for $e = (u, v) \in E(K)$ such that

$$g(u, v) = \frac{c(u, v)}{\mathcal{D} - \mathcal{D}(u, v) - \mathcal{D}(s, u)},$$

where \mathcal{D} is the maximum delay acceptable in the MRP instance, $\mathcal{D}(u, v)$ is the delay incurred in the path from u to v , and similarly, $\mathcal{D}(s, u)$ is the delay incurred in the path from s to u .

The described algorithm runs in $O(2^{\log \mathcal{D}})$, where \mathcal{D} is the maximum delay acceptable in the MRP instance. This is not polynomial, since \mathcal{D} can be very large. However, in practice the value of \mathcal{D} is small, and the algorithm behaves in most cases as a polynomial algorithm. The following interesting result provides a bound on the worst case performance of the described procedure.

Theorem 4. *The worst case of the algorithm described above is $l|D|/(l+|D|)$, for a given value l corresponding to edge costs. Thus, the worst case performance is asymptotically equal to $|D|$.*

3.2.1. *Sparsity considerations and the MRP.* Sparsity in the definition of multicast network instances is a major factor that must be considered in the design of new algorithms for the MRP. This is necessary since sparse instances can usually be handled by faster algorithms. As an example of application of this strategy, the heuristic proposed by Chung et al. [1997] was designed to provide solutions to the delay constrained MRP, when the instances are sparse. To do this, the algorithm employs a Lagrangian relaxation technique, implicitly used on an algorithm that was initially proposed by Blokh and Gutin [1996] (the algorithm is from now on called the BG heuristic). The BG heuristic solves a minimization problem with a constraint by recasting it as an unconstrained problem with Lagrangian multipliers. The multipliers are used in this context to force feasibility of the resulting solution.

The algorithm proposed in Chung et al. [1997] tries to adapt the BG heuristic to the MRP. To do this, it uses a well known algorithm for the Steiner problem [Takahashi and Matsuyama, 1980] as the generator for an initial solution. Then, it applies the scheme used by the BG heuristic to improve the initial solution, until a feasible solution for the MRP is found. According to empirical results in Chung et al. [1997], the heuristic provides results for the MRP that have on average 11% larger objective cost than the optimum solution.

3.2.2. *Multicast routing as an on-line problem.* A frequent problem that occurs when solving the MRP consists of dealing with reconfigurations of routes, when inclusions and exclusion of members of a group are allowed [Aguilar et al., 1986, Waxman, 1988]. In this case, although the initial tree may have been optimal according to some criterion, it is easy to see that after several insertions and deletions this may not be an optimal tree, and in fact it can be far from optimality. Thus, a major challenge on dynamic multicast applications is to allow for periodic reconfiguration of routes, with the objective of avoiding excessive degradation of solution quality.

To solve this major problem, some algorithms have been presented in the last few years [Kheong et al., 2001, Waxman, 1988]. A technique that was proposed by Waxman [1988] consists of maintaining caches with precomputed sub-trees that connect subsets of the destinations. Such “routing caches” can be used in two ways:

- First, caches can be used to store information about frequent combinations of nodes in a multicast group. In this sense, insertions and deletion of nodes can be used to “learn” new configurations of destinations, and how they are best satisfied, storing the resulting information in a routing tree database.
- Second, the cached information can be used to speed up the creation of multicast routing trees in the case of dynamic changes. A cache can also be used to quickly reconstruct routing trees, starting from parts of existing tree, which are known to be able to connect nodes that are currently part of the multicast group.

Waxman [1988] was one of the first researchers to consider heuristics for cost minimization taking in consideration the dynamics of inclusion and exclusion of members in a multicast group. The algorithm by Waxman [1988] presents one of the possible techniques to manage this database of routing subtrees. To do this, it provides a mathematical framework for updating trees formed by the dynamic inclusion and exclusion of nodes. He proposed a random graph model, where the probability of adding an edge between two nodes depends on the Euclidean distance between them. This probability decreases with distance between nodes. Using this model, a dynamic update rule is derived, which gives an improved way of constructing routes.

Other examples of algorithmic techniques for the on-line MRP are the following. The algorithm by Hong et al. [1998], which is in most aspects similar to Chung et al. [1997], is an implementation of dynamic resource management, and uses a strategy capable of handling additions and removals of elements to an existing multicast group. The algorithm is again based on the KMB algorithm for the Steiner tree problem. To reduce the complexity of the problem, the authors employed a Lagrangian relaxation technique.

Feng and Yum [1999] also devised a heuristic for the online version of the MRP, with the main goal of allowing easy insertion of new nodes in the multicast group. Similarly to the TM heuristic for the Steiner tree problem [Takahashi and Matsuyama, 1980], the proposed algorithm uses a strategy in which, at each step, it takes a non-connected destination with minimum cost and tries to add this destination to the current solution (this is similar to the way the Prim’s algorithm computes a solution to the minimum spanning tree problem).

Sriram et al. [1999] proposed new algorithms for the on-line, delay constrained minimum cost multicast routing which tries to maintain a fixed quality of service by specifying minimum delays. The algorithm is able to adapt the routing tree to changes in membership due to inclusions and exclusions of users.

Finally, a technique that has been explored in some algorithms for the online version of the MRP consists of using information available from unicast protocols. By doing this, the algorithm has two advantages:

- To simplify the main algorithm since it is not necessary to consider the lower level task of finding a single path — the objective becomes to integrate unicast paths in an optimal way;
- To reduce the computational time of the algorithm, by reusing information that was previously computed and left available by the unicast protocol.

An example of the use of this technique is given by Baoxian et al. [2000], who proposed a heuristic where paths are found by requesting this information to the underlying unicast protocol.

3.3. Distributed algorithms. Multicast routing is a task that depends essentially on the use of distributed algorithms. This is true since the MRP is defined on a distributed environment, where clients and servers are able to perform computation. This available computational power can be used to reduce the time complexity of the employed algorithms, through careful parallel implementation. A number of papers have focused on implementation of better distributed strategies for MRP problem, as discussed below [Jia, 1998, Chen et al., 1993].

A good example of distributed algorithm for the MRP is provided by Kompella et al. [1993a]. The proposed algorithm was targeted at applications where audio and video need to be delivered over a network, and where delay restrictions play an important role. In the first algorithm presented in Kompella et al. [1993a], the Bellman-Ford algorithm [Bellman, 1957] (which is used to compute the shortest path between two nodes in a network) is extended with the objective of finding a minimum delay tree from the source to each destination. A second algorithm was also proposed, with similar techniques, but this time using a strategy similar to the Prim's algorithm for minimum spanning tree construction.

Chen et al. [1993] have further studied distributed algorithms to the MRP. The authors proposed a heuristic that is similar to the algorithm described by Kompella et al. [1993b] (and also to the technique used in the KMB heuristic for Steiner tree). The main difference in this case, however, is that a distributed algorithm has been employed to find a minimum spanning tree, which must be computed twice during the execution of the heuristic. The method used to find the MST is based on the distributed algorithm proposed by Gallager et al. [1983]. The use of this distributed algorithm allowed a considerable reduction on computational time, providing an efficient implementation.

Shaikh and Shin [1997] have presented a distributed algorithm where the focus is to reduce the complexity of distributed versions of heuristics for the the delay constrained Steiner problem. In their paper, the authors try to adapt the general model of Prim's and Dijkstra's algorithms to the harder task of creating a multicast routing tree. The main addition done to the structure of Dijkstra's algorithm is a method to distinguish between destinations and non destination nodes, as presented in Algorithm 3. Here, function $I_D : V \rightarrow \{0, 1\}$ is an indicator function, such that if $v \in D$, then $I_D(v) = 0$, with $I_D(v) = 1$ otherwise. The objective of this indicator function is to remove the costs of adding a destination to the current tree, making it easier to construct a tree linking the source to destinations.

3.3.1. Algorithms for sparse groups. An important case of the MRP occurs when the number of sources and destinations is small compared to the whole network. This is the typical case for large instances, where just a few nodes will participate in a group, at each moment. To handle sparsity in the MRP, Sriram et al. [1998] proposed a distributed algorithm which tries to explore properties of sparse instances. The algorithm uses the unicast routing algorithms that already exist in the underlying network. The algorithm is composed of two

```

procedure Modified-Dijkstra
Input:  $G(V, E), s$ 
begin
  for  $v \in V$  do  $d[v] \leftarrow \infty$ ;
   $d[s] \leftarrow 0$ ;
   $S \leftarrow \emptyset$ ;
   $Q \leftarrow V$  /*  $Q$  is a queue */;
  repeat
     $v \leftarrow \text{get\_min}(Q)$ ;
     $Q \leftarrow Q \setminus \{v\}$ ;
     $S \leftarrow S \cup \{v\}$ ;
    foreach  $u \in N(v)$  do
      if  $u \notin S$  and  $d[u] > d[v] + w(u, v)$  then
         $d[u] \leftarrow d[v] + w(u, v)$ ;
      end
    until  $Q$  is empty;
end

```

Algorithm 3: Modification of Dijkstra’s algorithm for multicast routing, as proposed by Shaikh and Shin [1997].

phases: in the first one, paths are computed from the source to each destination, using the underlying unicast protocol. This phase of the algorithm can be performed in a distributed way, since the unicast protocols are also distributed. In the second phase, the algorithm has to apply some heuristic rules to define what segments of the returned paths will be used as part of the final routing tree. Some rules have been proposed on Sriram et al. [1998], depending on the configuration of the possible intersections between the given paths.

A problem that arises when multicast groups are allowed to have dynamic membership is that a considerable amount of time is spent in the process of connection configuration. Jia [1998] proposed a distributed algorithm which addresses this question. The main assumptions used by the algorithm are the following:

- *Minimizing computational effort.* Most of the computational effort should be spent by nodes that participate in the multicast group (source and destination), other nodes should not be heavily penalized.
- *Decentralized management of routing information.* Routing information must be distributed, and every node should be capable of collecting its own routing information. This allows the algorithm to operate independently of any central node that stores the global state of the system.
- *Minimizing number of messages.* The algorithm should use the least possible amount of messages, and each message should have a small size.

Using these principles, the algorithm applies the following steps. Starting from the source node s , all the shortest paths from the source to the destinations are computed. To do this, it is possible to use information available from the unicast protocol. The first step consists of linking s to the destination $v \in D$ that is closest to s . Then, in the following iterations, shortest paths are computed from the subtree T that has been constructed in the previous steps; the node $v \in D$ that is closest to one of the nodes $w \in T$ is selected, and the path \mathcal{P} connecting them is added to the tree. Finally, redundant parts of the union $T \cup \mathcal{P}$ are removed, in such a way that delay constraints are satisfied.

3.4. Other restrictions: Nonsymmetric links, degrees, and delay variation. An interesting feature of real networks, which is not mentioned in most research papers, is that links are, in general, nonsymmetric. The capacity in one direction can be different from the capacity in the other direction, for example, due to congestion problems in some links. Ramanathan [1996] considered this kind of restriction, and proposed an approximation algorithm with constant worst case guarantee. The resulting algorithm has also the nice characteristic of being parameterizable, and therefore it allows the trading of execution time for accuracy.

If the number of links from any node in the network is required to be a fixed value, then we have the *degree-constrained* version of the multicast routing problem. As an example where this is an important feature we can cite ATM networks. In this type of application, the number of out-connections must have a fixed limit, dictated by the type of routers used in the system [Zhong et al., 1993]. A combinatorial algorithm for the MRP with maximum degree constraints was provided by Bauer and Varma [1995] and Bauer [1996]. In such algorithms, the basic strategy of selecting Steiner trees with the required nodes is modified in a way such that only trees with limited number of connections per node are allowed.

Another restriction that is frequently disregarded in most algorithms is related to the so called *delay variation*. The delay variation is defined as the difference between the minimum and maximum delays between source and destinations, as defined by a specific routing tree. In some applications, such as video-conferencing, virtual reality, and other collaborative environments, it is interesting that the delay variation stay within a specific range. For example, it may be desirable that all nodes receive the same information at about the same time, to improve synchrony among participants of the multicast group. The delay variation constraint was considered by Rouskas and Baldine [1996], who proposed an algorithm for its minimization.

3.5. Center based routing. An important technique for routing tree construction is the so called *center based algorithms* [Salama et al., 1996, Wall, 1982]. The objective of center based algorithms is to find a tree rooted on some node of the original graph G , with a specific centrality property. As an example, we can consider using as root a node r that minimizes the maximum distance to all other nodes in G . This idea has been pursued in some algorithms for the MRP, and is implemented in protocols such as CBT (core-based tree) [Ballardie et al., 1993].

An example of application of center based algorithms occurs when a routing tree must be shared among several multicast groups (a technique that is called shared tree construction). In this case, it is interesting to use a central node that is as close as possible from all other sources and destinations.

Another example of center based routing usage is in finding the topological center of a set of nodes. The *topological center* is the node that is closest to all members of the involved multicast groups, and is known to be an NP-hard problem [Ballardie et al., 1993]. In other words, the objective is to find a node $v \in V$ that is closest to any other node in the network, i.e., the node v minimizing $\max_{u \in V} d(v, u)$. A routing tree rooted at v is then constructed and used throughout the multicast session – the idea is that such tree will be more “balanced” and therefore will need less changes as nodes are added or removed from the original multicast group(s).

To avoid the complexity problems involved with the concept of topological center, other centrality measures have been used in practical protocol implementations. Examples of alternative centrality measures are *core points* [Ballardie et al., 1993] and *rendez-vous*

points [Deering et al., 1994]. A comparison of different methods for computing a center for a routing tree is provided by Calvert et al. [1995].

3.6. Delay constrained minimum spanning tree problem. Finally, an alternative solution method for the computation of routes in multicast systems consists of constructing a spanning tree problem with delay constraints. This problem, similarly to the shortest path problem with constraints, is NP-hard [Garey and Johnson, 1979]. Therefore, heuristics are used in practice to provide solutions in polynomial time. An example of heuristic solution was proposed by Chow [1991], using the strategy of combining different routes into one single routing tree. Another heuristic algorithm for this problem was provided by Salama et al. [1997a]. This heuristic is a modified version of the Prim's algorithm, implemented using distributed computation techniques. A comparison of the algorithms for this problem can also be found in Salama et al. [1997b].

4. THE MULTICAST PACKING PROBLEM

Capacity planning for the necessary bandwidth in a multicast network is another important problem that occurs during the implementation phase of multicast systems. The main requirement when considering this problem is that the network links must have enough capacity to satisfy all multicast groups sharing the network. Similarly, when the maximum capacity is limited, it must be shared by all groups according to their bandwidth requirements. These capacity constraints are modeled in what is called the *multicast packing problem*. This problem has attracted considerable attention from researchers in the area of network design in the past few years [Wang et al., 2002, Priwan et al., 1995, Chen et al., 1998].

In this section, we present a formulation for the multicast packing problem, along with some of the computational techniques that have been applied for its solution.

Given a graph $G = (V, E)$ representing a network, the congestion λ_e on edge $e \in E(G)$ is defined as the sum of all load imposed by the multicast groups using e . With this notation, the maximum congestion λ is defined as the maximum congestion λ_e , taken over all edges $e \in E$. If we assume that there are K multicast groups, and each group k generates an amount t^k of traffic, an integer program for the multicast packing problem is given by

$$(8) \quad \min \lambda$$

subject to

$$(9) \quad \sum_{i=1}^K t^i x_e^i \leq \lambda \quad \text{for all } e \in E$$

$$(10) \quad x_e^k \in \{0, 1\}^{|E|} \quad \text{for } i \in \{1, \dots, K\},$$

where variable x_e^k is equal to one if and only if the edge e is being used by multicast group k . Note that the variables x_e^k must be themselves determined using an integer programming formulation, as the one presented in the previous section.

The multicast packing problem formulated above is NP-hard, and as such it can only be approximately solved by polynomial time algorithms. Heuristic strategies for the problem have been proposed by Wang et al. [2002]. The authors discuss two main algorithms to set up multiple multicast groups, and formalized this as a packing problem.

The first heuristic uses a Steiner tree-based strategy — solutions are created in a greedy way, with edges being selected to the Steiner tree if they have enough capacity to satisfy the requirements of the multiple multicast groups. The proposed algorithm has been shown

to run in time $O(kn^4 + k^2mn^2)$, where k is the number of multicast groups, n the number of nodes, and m the number of links.

The second heuristic is based on the idea of computing the cut-sets between pairs of elements of the multicast group. By determining the cut-sets for each such pair of nodes, one can compute the edges that need to be part of a routing tree. The proposed algorithm repeats the tree computation for each multicast group, using each time the *residual graph* (i.e., the graph representing the remaining capacity at a specific step).

4.1. Alternative formulations for the multicast packing problem. Alternative formulations for packing problems in multicast networks were presented by Chen et al. [1998], using two integer programming models. Initially, suppose we are given a graph $G = (V, E)$ and a set of costs w_e , for each edge $e \in E(G)$. In their first integer formulation of the multicast packing problem, Chen et al. [1998] used a binary 0-1 variable x_e for each edge $e \in E$. The variable x_e is equal to one if and only if the edge e is selected, with respective cost w_e . Then the integer formulation for the tree version of the problem is given by

$$(11) \quad \min \sum_{e \in E} w_e x_e$$

subject to

$$(12) \quad \sum_{e \in \delta(S)} x_e \geq 1 \quad \text{for all } S \subset V \text{ such that } m_1 \in S \text{ and } M \not\subset S$$

$$(13) \quad x \in \{0, 1\}^{|E|},$$

where M is the set of nodes participating in a multicast group, node m_1 is an arbitrary member of M , and $\delta(S)$ represents the edges leaving the set $S \subset V$. This is basically the same formulation used for the Steiner tree problems, and can be solved using similar techniques, including branch-and-cut algorithms based on the known cuts for the Steiner problem.

In a second IP formulation to the multicast packing problem, the variables x_e , for $e \in E(G)$, are similar to the ones introduced above. The idea, however, is to use a *ring-based* version of the problem. In the ring-based multicast packing, the routes are defined as a ring (instead of a tree) linking the elements of the multicast group. The ring-based configuration is preferred in some situations, since it provides better reliability in the case of failures of a single link. The IP formulation for this version of the problem is given by

$$(14) \quad \min \sum_{e \in E} w_e x_e$$

subject to

$$(15) \quad \sum_{e \in \delta(v)} x_e = 2 \quad \text{for all } v \in M$$

$$(16) \quad \sum_{e \in \delta(v)} x_e \leq 2 \quad \text{for all } v \in V \setminus M$$

$$(17) \quad \sum_{e \in \delta(S)} x_e \geq 2 \quad \text{for all } S \subset V, u \in S, \text{ and } M \not\subset S$$

$$(18) \quad x \in \{0, 1\}^{|E|}.$$

Here, u is an arbitrary (but fixed) element of M . The integer solution of this formulation gives a ring passing through all nodes participating in group M . To solve this problem, one can also employ a branch-and-cut algorithm. In Chen et al. [1998], some valid inequalities for this problem have been used to implement such a branch-and-cut.

4.2. The multicast network dimensioning problem. Another interesting problem occurs when we consider the design of a new network, intended to support a specific multicast demand. This is called the *multicast network dimensioning problem*, and it has been studied in some recent papers [Prytz, 2002, Forsgren and Prytz, 2002, Prytz and Forsgren, 2002].

The problem consists of determining the topology (i.e., which edges need to be selected) and the corresponding capacity of the edges, such that a specific multicast service can be deployed in the resulting network. Much of the work related to this problem has used mathematical programming techniques to provide exact and approximate solutions.

The following formulation was derived by Forsgren and Prytz [2002]. In that paper, the technique used has been the Lagrangian relaxation applied to an integer programming model. We assume in this formulation a network represented by the graph $G = (V, E)$ and costs w_e , for $e \in E(G)$. Moreover, we assume there are T multicast groups. The model uses variables $x_e^k \in \{0, 1\}$, for $k \in \{1, \dots, T\}$, and $e \in E$, which indicate if edge e is used by group k . There are also variables $z_e^l \in \{0, 1\}$, for $l \in \{1, \dots, L\}$, and $e \in E$ (where L is the highest possible capacity level), which determine if the capacity level of edge e is equal to l . Now, let d_k , for $k \in \{1, \dots, T\}$, be the bandwidth demanded by group k . Also, let c_e^l , for $l \in \{1, \dots, L\}$, and $e \in E$, be the capacity available for edge e at the level l ; and w_e^l , for $l \in \{1, \dots, L\}$, and $e \in E$, be the cost of using edge e at the capacity level l . Assume that $b \in \mathbb{Z}^n$ is the demand vector, and $A \in \mathbb{R}^{n \times m}$ is the node-edge incidence matrix. We can now state the multicast network dimensioning problem using the following integer program

$$\begin{aligned}
 (19) \quad & \min && \sum_{e \in E} \sum_{l=1}^L w_e^l z_e^l \\
 (20) \quad & \text{subject to} && \sum_{k=1}^T d^k x_e^k \leq \sum_{l=1}^L c_e^l z_e^l \quad \text{for all } e \in E \\
 (21) \quad & && \sum_{l \in L} z_e^l \leq 1 \quad \text{for all } e \in E \\
 (22) \quad & && Ax = b \\
 (23) \quad & && x, z \text{ integral.}
 \end{aligned}$$

In this integer program, constraint (20) ensures that the bandwidth used on each edges is at most the available capacity. constraint (21) selects just one capacity level for each edge. Finally, constraint (22) enforces the flow conservation in the resulting solution. Instances of the problem proposed above have been solved using a branch-and-cut algorithm. Valid inequalities proved in Forsgren and Prytz [2002] have been used to derive a class of cuts, which were then implemented in the branch-and-cut framework.

5. OTHER PROBLEMS IN MULTICAST ROUTING

5.1. The point-to-point connection problem. The point-to-point connection problem (PPCP) is a generalization of the Steiner problem that can be used to model multicast systems with multiple sources and destinations. In the PPCP, we are given a graph $G = (V, E)$ with cost function $w : E \rightarrow \mathbb{R}$ on the edges, a set of sources $S \subset V$, and a set of destination $D \subset V$ such that $|V| = |D|$ and $V \cap D = \emptyset$. The objective of the problem is to find a forest (a set of o trees) $F \subseteq E$ connecting sources to destinations, such that there is a bijective mapping $\phi : V \rightarrow V$ between elements of S and elements of D , and F has minimum cost. If the map ϕ is fixed in advance, then this is known as the *fixed-destinations version* of the PPCP.

The PPCP can be thought as a general version of the MRP, where multiple sources are allowed. The requirement that the number of sources be equal to the number of destinations can be satisfied by any PPCP instance with the addition of “dummy” sources, without loss of generality. Thus, solving the PPCP is important when the multicast structure is distributed, and sources are replicated over the network.

The PPCP has been proved to be NP-hard [Li et al., 1992], considering its directed and undirected versions, and even for the fixed-destinations version. For the especial case with $p = 2$ (where p is the number of destinations), an algorithm with time complexity $O(n^5)$ was also proposed. Later, Natu [1995] improved on this case of the problem, with a dynamic programming algorithm with time complexity $O(mn + n^2 \log n)$. The same authors proposed an algorithm for the case $p = 3$ with time complexity $O(n^{11})$.

In terms of approximation complexity, it is known that the PPCP can be approximated within a constant ratio. In their seminal work, Goemans and Williamson [1995] presented approximation algorithms for a large class of forest constrained problems, including the PPCP and the Steiner problem. The approximation algorithm, when applied to the PPCP, runs in $O(n^2 \log n)$ time, and gives its results within a factor $2 - 1/p$ of the optimal solution.

Given the inherent complexity of the PPCP, finding optimal solutions requires the use of clever enumeration schemes that try to avoid the search through all possible solutions. A branch-and-bound algorithm along these lines was proposed by Meneses et al. [2004]. To use such an approach, one needs to state the problem as an integer program. Let x_e be a binary variable equal to 1 if and only if the edge $e \in E(G)$ is selected to be part of the solution. Then, we can write the PPCP as the following problem:

$$(24) \quad \begin{aligned} & \min \sum_{e \in E(G)} c_e x_e \\ & \text{subject to } \sum_{e \in \delta(A)} x_e \geq 1 \quad \text{for all } A \subset V \text{ with } |A \cap S| \neq |A \cap D| \\ & \quad x_e \in \{0, 1\}. \end{aligned}$$

An important property of this formulation is summarized in the the following theorem.

Theorem 5. *The inequality*

$$\sum_{e \in \delta(A)} x_e \geq 1 \quad \text{for all } A \subset V \text{ with } |A \cap S| \neq |A \cap D|$$

is a valid inequality, i.e., it must be satisfied by any feasible solutions $x \in \{0, 1\}^E$ for the PPCP.

The valid inequality (24) can be used to provide a branch-and-cut algorithm for the PPCP. The basic strategy consists of adding only a small number of inequalities of the type (24), and then check if the solution of the resulting formulation is optimal. We can check if a solution does not violate any inequalities of the type (24) using the following algorithm: given a solution x to the integer program above, create a network with underlying graph $G = (V, E)$ and capacities $cap(e) = x_e$, for $e \in E(G)$. For each combination $v_s \in S$, $v_t \in D$ of source-destination pairs, run a maximum flow algorithm from v_s to v_t . If the maximum flow between any two source-destination pair has value less than one, this means that one of the inequalities (24) has not been satisfied. It is then easy to identify such inequality, since it corresponds to the minimum cut corresponding to the maximum flow computed previously.

Heuristic approaches for the PPCP have been proposed by Correa et al. [2003] and Gomes et al. [1998]. The technique used for solving the problem relies on the cooperation of multiple algorithms, also called agents, that manipulate a population of feasible solutions for the problem. The technique, called *asynchronous teams* (A-Teams) combines several agents to optimize an objective function over the set of feasible solutions for a combinatorial problem [Talukdar and de Souza, 1990]. An asynchronous teams strategy for the PPCP was proposed in Gomes et al. [1998], with results close to the optimal for most of the tested instances. The asynchronous nature of the A-Teams metaheuristic makes it very natural to use parallel and distributed algorithms to implement it. A parallel implementation of the A-Teams for the PPCP was proposed by Correa et al. [2003], showing a clear improvement of solution values with increasing number of processors.

5.2. Streaming cache placement problems in multicast routing. Although multicasting is an important operation in modern networks, most of the current routers using the TCP/IP protocol do not support multicasting by default. Therefore, the economical implementation of multicast systems is a practical issue, and one of the challenges in the design of multicast systems.

One of the models for the design of economic multicast networks is the *cache placement problem* (CPP). The main issue addressed by the CPP is that of restricting the number of replication nodes, also known as *cache nodes*. Notice however, that due to capacity constraints, a minimum number of cache nodes is required to simultaneously transfer data from the source to all destinations. The objective of the CPP is to find the minimum number of nodes necessary to satisfy all demand, given the capacity constraints.

Using a graph model to describe the CPP, we are given a graph $G = (V, E)$ with capacities c_e on each edge $e \in E$, a source node s and a set of destinations $D \subset V(G)$. Then, the objective of the CPP consists of finding a set R of replication nodes, also known as *cache nodes*, such that all destinations receive the required data, and all capacity constraints are satisfied. To better describe the requirements of the CPP, we discuss the following integer programming model of the problem. Let the integer variables y , x , b , and w be described as

$$y_e = \begin{cases} 1 & \text{if edge } e \text{ is in the spanning tree } T \\ 0 & \text{otherwise,} \end{cases}$$

$$x_i = \begin{cases} 1 & \text{if node } i \neq s \text{ is a cache node} \\ 0 & \text{otherwise} \end{cases}$$

$$b_i \in \{-1, \dots, |V| - 1\} \quad \text{the flow surplus for node } i \in V$$

$$w_e \in \{0, \dots, |V|\} \quad \text{the amount of flow in edge } e \in E.$$

Where A is the node-arc incidence matrix corresponding to the graph G . The problem can now be stated as

$$(25) \quad \min \sum_{i=1}^{|V|} x_i$$

subject to

$$(26) \quad Aw = b$$

$$(27) \quad \sum_{i \in V} b_i = 0$$

$$(28) \quad b_s \geq 1 \quad \text{for source } s$$

$$(29) \quad x_i - 1 \leq b_i \leq x_i(|V| - 1) - 1 \quad \text{for } i \in D$$

$$(30) \quad x_i \leq b_i \leq x_i(|V| - 1) \quad \text{for } i \in V - (D \cup \{s\})$$

$$(31) \quad \sum_{e \in E} y_e = |V| - 1$$

$$(32) \quad \sum_{e \in G(H)} y_e \leq |H| - 1 \quad \text{for all } H \subset V$$

$$(33) \quad 0 \leq w_e \leq c_e y_e \quad \text{for } e \in E$$

$$(34) \quad x \in \{0, 1\}^{|V|}, \quad y \in \{0, 1\}^{|E|}$$

$$(35) \quad b \in Z, \quad w \in Z^+,$$

where $G(H)$ is the subgraph induced by the nodes in H .

Recall that in the CPP, as in the MRP, a feasible solution is a tree, rooted at the source node s , connecting s to the destinations. Another version of the CPP arises when we relax the constraints (31) and (32), which define the solution as a tree on G . In this case, we have a problem where the objective is to send data from source to destinations using any feasible flow, not restricted to a tree. The resulting problem is called the *flow version* of the CPP. Notice that this makes sense in the CPP because only a small part of the nodes will be elements of R , and other nodes can receive flow according to multiple configurations that are not necessarily a tree.

Despite the importance of the CPP for the economical design of multicast networks, only recently it has been studied; an initial result shows that the problem is NP-hard, as stated in the following theorem [Mao et al., 2003, Oliveira et al., 2003, Mao et al., 2003]:

Theorem 6. *The CPP in the tree-version ((25) to (35)) is NP-hard. Moreover the result is true for the flow version of the CPP, as well as the cases where the underlying graph is directed or undirected.*

Possible techniques for solving the CPP include heuristic, and enumerative methods. Heuristic algorithms are important for this problem, since it is difficult to find approximation algorithms with good approximation guarantee. In fact, it has been shown in Oliveira and Pardalos [2005], using a reduction from SET COVER [Feige, 1998], that the problem cannot be approximated by a factor better than $\log |D|$, unless NP is easy to solve:

Theorem 7. *If there is some $\epsilon > 0$ such that a polynomial time algorithm A can approximate CPP within $(1 - \epsilon) \log k$, where $k = |D|$, then $NP \subset TIME(n^{O(\log \log n)})$.*

The theorem above shows that no algorithm can guarantee a good solution for the CPP problem in general. Nonetheless, heuristics proposed in Oliveira and Pardalos [2005] were demonstrated to return near optimal solutions to the CPP for most instances. It is an open problem to find algorithms for the CPP that can match the approximation lower bound given above — while the techniques developed for SET COVER appear to be a good starting point, they do not seem to be directly applicable in solving the CPP.

6. CONCLUDING REMARKS

In this chapter, we discussed several optimization problems occurring in multicast networks. The rich combinatorial structure of multicasting makes it possible to provide several perspectives of the problem, according to the different and sometimes unrelated objectives that must be optimized, as well as the various constraints that must be satisfied by practical systems.

Most of the problems concerning multicast routing present interesting challenges for researchers working in combinatorial optimization. First, the problems in this area are of practical interest, since most network applications could benefit from good algorithms for the discussed problems. Second, many of these problems have not been fully explored using tools of optimization. As shown in this chapter, most algorithms are only concerned with the generation of feasible solutions, with some additional requirements used to maintain a minimum quality of service. It remains an open question how the solutions returned by the discussed algorithms can be improved by the application of optimization techniques, such as local search optimization, metaheuristics, branch-and-cut/branch-and-price, and approximation algorithms. Therefore, we believe that important advances can be made in the near future by the application of optimization techniques to problems in multicast routing.

REFERENCES

- L. Aguilar, J.J. Garcia-Luna-Aceves, D. Moran, E.J. Graighill, and R. Brungardt. Architecture for a multimedia teleconferencing system. In *Proceedings of the ACM SIGCOMM*, pages 126–136, Baltimore, Maryland, 1986. Association for Computing Machinery.
- M. Baldi, Y. Ofek, and B. Yener. Adaptive real-time group multicast. In *Proceedings of IEEE INFOCOM'97*, page 683, 1997.
- A. Ballardie, P. Francis, and J. Crowcroft. Core-based trees (CBT) – An architecture for scalable inter-domain multicast routing. *Computer Communication Review*, 23(4):85–95, 1993.
- Z. Baoxian, L. Yue, and C. Changjia. An efficient delay-constrained multicast routing algorithm. In *International Conference on Communication Technologies (ICCT 2000)*, page S07.2, 2000.
- F. Bauer. *Multicast routing in point-to-point networks under constraints*. PhD thesis, University of California, Santa Cruz, 1996.
- F. Bauer and A. Varma. Degree-constrained multicasting in point-to-point networks. In *Proceedings IEEE INFOCOM '95, The Conference on Computer Communications*, pages 369–376, 1995.
- F. Bauer and A. Varma. ARIES: A rearrangeable inexpensive edge-based on-line Steiner algorithm. *IEEE Journal of Selected Areas in Communications*, 15(3):382–397, 1997.
- M. Bazaraa, J. Jarvis, and H. Sherali. *Linear Programming and Network Flows*. John Wiley and Sons, 2nd edition, 1990.
- R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- L.T.M. Berry. Graph theoretic models for multicast communications. *Computer Networks and ISDN Systems*, 20(1):95–99, 1990.
- K. Bharath-Kumar and J.M. Jaffe. Routing to multiple destinations in computer networks. *IEEE Transactions on Communications*, 31(3):343–351, 1983.
- D. Blokh and G. Gutin. An approximate algorithm for combinatorial optimization problems with two parameters. *Australasian J. Combin.*, 14:157–164, 1996.

- K.L. Calvert, E.W. Zegura, and M.J. Donahoo. Core selection methods for multicast routing. In *IEEE ICCCN'95*, pages 638–642, Las Vegas, Nevada, 1995. IEEE.
- K. Carlberg and J. Crowcroft. Building shared trees using a one-to-many joining mechanism. *ACM Computer Communication Review*, 27(1):5–11, January 1997.
- G. Chen, M. Houle, and M. Kuo. The Steiner problem in distributed computing systems. *Information Sciences*, 74(1):73–96, 1993.
- S. Chen, O. Günlük, and B. Yener. Optimal packing of group multicasting. In *Proc. IEEE INFOCOM'98*, pages 980–987, 1998.
- C. Chiang, M. Gerla, and L. Zhang. Adaptive shared tree multicast in mobile wireless networks. In *Proceedings of GLOBECOM'98*, pages 1817–1822, 1998.
- G.V. Chockler, N. Huleihel, I. Keidar, and D. Dolev. Multimedia multicast transport service for groupware. In *TINA Conference on the Convergence of Telecommunications and Distributed Computing Technologies*, pages 43–54, 1996.
- C.H. Chow. On multicast path finding algorithms. In *Proc. IEEE INFOCOMM'91*, pages 1274–1283, 1991.
- S.-J. Chung, S.-P. Hong, and H.-S. Huh. A fast multicast routing algorithm for delay-sensitive applications. In *IEEE GLOBECOM'97*, pages 1898–1902, 1997.
- R. Correa, F. Gomes, C.A.S. Oliveira, and P.M. Pardalos. A parallel implementation of an asynchronous team to the point-to-point connection problem. *Parallel Computing*, 29(4):447–466, 2003.
- Y. Dalal and R. Metcalfe. Reverse path forwarding of broadcast packets. *Communications of the ACM*, 21(12), 1978.
- S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended LANs. *ACM Transactions on Computer Systems*, 8(2):85–110, 1990.
- S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei. An architecture for wide-area multicast routing. *Computer Communication Review*, 24(4):126–135, 1994.
- S. Deering, D.L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei. The PIM architecture for wide-area multicast routing. *IEEE/ACM Transactions on Networking*, 4(2):153–162, 1996.
- E.W. Dijkstra. A note on two problems in connexion with graphs. *Numer. Math.*, 1:269–271, 1959.
- M. Doar and I. Leslie. How bad is naive multicast routing. In *Proceedings of the IEEE INFOCOM*, pages 82–89, Los Alamitos, Calif, USA, 1993. IEEE Comput. Soc. Press.
- D.-Z. Du, B. Lu, H. Ngo, and P.M. Pardalos. Steiner tree problems. In C.A. Floudas and P.M. Pardalos, editors, *Encyclopedia of Optimization*, volume 5, pages 227–290. Kluwer Academic Publishers, 2001.
- D.-Z. Du and P.M. Pardalos, editors. *Network Optimization Problems: Algorithms, Complexity and Applications*. World Scientific, 1993a.
- D.-Z. Du and P.M. Pardalos. Subset interconnection designs: Generalizations of spanning trees and Steiner trees. In *Network Optimization Problems*, pages 111–124. World Scientific, 1993b.
- C.A. Ellis, S.J. Gibbs, and G.L. Rein. Groupware: Some issues and experiences. *Commun. ACM*, 34(1):39–58, 1991.
- H. Eriksson. MBONE – the multicast backbone. *Communications of ACM*, 37(8):54–60, 1994.
- U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

- G. Feng and T. Peter Yum. Efficient multicast routing with delay constraints. *International Journal of Communication Systems*, 12:181–195, 1999.
- A. Forsgren and M. Prytz. Dimensioning multicast-enabled communications networks. *Networks*, 39:216–231, 2002.
- A.J. Frank, L.D. Wittie, and A.J. Bernstein. Multicast communication on network computers. *IEEE Software*, 2(3):49–61, 1985.
- R.G. Gallager, P.A. Humblet, and P.M. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Trans. Programming Languages and Systems*, 5(1):66–77, 1983.
- M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W.H. Freeman and Company, 1979.
- M.X. Goemans and D.P. Williamson. A General Approximation Technique for Constrained Forest Problems. *SIAM J. Comp.*, 24(2):296–317, 1995.
- F.C. Gomes, C.N. Meneses, A.R.G. Lima, and C.A.S. Oliveira. Asynchronous organizations for solving the point-to-point connection problem. In *Proc. of the Intl. Conference on Multiagents Systems (ICMAS)*, pages 144–149. IEEE Computer Society, 1998.
- L. Han and N. Shahmehri. Secure multicast software delivery. In *IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'00)*, pages 207–212, 2000.
- S. Hong, H. Lee, and B.H. Park. An efficient multicast routing algorithm for delay-sensitive applications with dynamic membership. In *Proceedings of IEEE INFOCOM'98*, pages 1433–1440, 1998.
- F. Hwang and D. Richards. Steiner tree problems. *Networks*, 22:55–89, 1992.
- F. Hwang, D.S. Richards, and P. Winter. *The Steiner tree problem*, volume 53 of *Annals of Discrete Mathematics*. North-Holland, 1992.
- IBM. Reliable multicast messaging, 2005. Available at <http://www.haifa.il.ibm.com/projects/software/rmsdk>.
- Y. Im, Y. Lee, S. Wi, and Y. Choi. A delay constrained distributed multicast routing algorithm. *Computer Communications*, 20(1):60–66, 1997.
- X. Jia. A distributed algorithm of delay-bounded multicast routing for multimedia applications in wide area networks. *IEEE/ACM Transactions on Networking*, 6(6):828–837, 1998.
- X. Jia, N. Pissinou, and K. Makki. A real-time multicast routing algorithm for multimedia applications. *Computer Commun. J.*, 20(12):1098–1106, 1997.
- X. Jiang. Routing broadband multicast streams. *Computer Communications*, 15(1):45–51, 1992.
- Chee Kheong, David Siew, and Gang Feng. Efficient setup for multicast connections using tree-caching. In *Proceedings IEEE INFOCOM 2001*, pages 249–258, 2001.
- V. Kompella, J. Pasquale, and G. Polyzos. Multicasting for multimedia applications. In *Proceedings of IEEE INFOCOM'92*, pages 2078–2085, 1992.
- V. Kompella, J. Pasquale, and G. Polyzos. Two distributed algorithms for the constrained Steiner tree problem. In *Proceedings of the Second International Conference on Computer Communications and Networking (ICCCN'93)*, pages 343–349, 1993a.
- V.P. Kompella, J.C. Pasquale, and G.C. Polyzos. Multicast routing for multimedia communication. *IEEE/ACM Trans. Networking*, 1(3):286–292, 1993b.
- V.P. Kompella, J.C. Pasquale, and G.C. Polyzos. Optimal multicast routing with quality of service constraints. *Journal of Network and Systems Management*, 4(2):107–131, 1996.
- L. Kou, G. Markowsky, and L. Berman. A fast algorithm for Steiner trees. *Acta Informatica*, 15:141–145, 1981.

- C. Li, S. McCormick, and D. Simchi-Levi. The point-to-point delivery and connection problems: Complexity and algorithms. *Discrete Applied Math.*, 36:267–292, 1992.
- Z. Li and P. Mohapatra. QMBF: A QoS-aware multicast routing protocol. *Computer Communications Journal*, 26(6):611–621, 2003.
- Z.M. Mao, D. Johnson, O. Spatscheck, J.E. van der Merwe, and J. Wang. Efficient and robust streaming provisioning in VPNs. In *Proceedings of the WWW2003*, Budapest, Hungary, May 2003.
- C.N. Meneses, C.A.S. Oliveira, and P.M. Pardalos. A branch-and-cut to the point-to-point connection problem on multicast networks. In F. Giannessi and A. Maugeri, editors, *Variational Analysis and Applications*, page 1084. Kluwer Academic Publishers, Dordrecht, 2004.
- J. Moy. Multicast extensions to OSPF, RFC 1584 – IETF network working group. On-line document: <http://www.ietf.org/>, 1994a.
- J. Moy. OSPF version 2, RFC 1583 – IETF network working group. On-line document: <http://www.ietf.org/>, 1994b.
- M.G. Natu. *Network Loading and Connection Problems*. PhD thesis, Operations Research Dept., North Carolina State University, 1995.
- C.A.S. Oliveira and P.M. Pardalos. Construction algorithms and approximation bounds for the streaming cache placement problems in multicast networks. *Cybernetics and systems Analysis*, 41, 2005. To appear.
- C.A.S. Oliveira, P.M. Pardalos, O.A. Prokopyev, and M.G.C. Resende. Streaming cache placement problems: Complexity and algorithms. Technical report, University of Florida, 2003.
- P.M. Pardalos and D.-Z. Du, editors. *Network Design: Connectivity and Facilities Location*, volume 40 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1998.
- P.M. Pardalos, F. Hsu, and S. Rajasekaran, editors. *Mobile Networks and Computing*, volume 52 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 2000.
- P.M. Pardalos and B. Khoury. An exact branch and bound algorithm for the Steiner problem in graphs. In D.-Z. Du & M. Li, editor, *Proceedings of COCOON'95*, volume 959 of *Lecture Notes in Computer Science*, pages 582–590. Springer-Verlag, Xi'an, China, 1995.
- P.M. Pardalos and B. Khoury. A heuristic for the Steiner problem on graphs. *Comp. Opt. & Appl.*, 6:5–14, 1996.
- P.M. Pardalos, B.N. Khoury, and D.-Z. Du. A test problem generator for the steiner problem in graphs. *ACM Transactions on Mathematical Software*, 19(4):509–522, 1993.
- J. Park and C. Park. Development of a multi-user & multimedia game engine based on TCP/IP. In *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, pages 101–104, Victoria, B.C. Canada, 1997.
- J. Pasquale, G. Polyzos, and G. Xylomenos. The multimedia multicasting problem. *ACM Multimedia Systems Journal*, 6(1):43–59, 1998.
- P. Paul and S.V. Raghavan. Survey of multicast routing algorithms and protocols. In *Proceedings of the Fifteenth International Conference on Computer Communication (ICCC 2002)*, pages 902–926, 2002.
- V. Priwan, H. Aida, and T. Saito. The multicast tree based routing for the complete broadcast multipoint-to-multipoint communications. *IEICE Transactions on Communications*, E78-B(5):720–728, 1995.

- M. Prytz. *On optimization in Design of Telecommunications Networks with Multicast and Unicast Traffic*. PhD thesis, Dept. of Mathematics, Royal Institute of Technology, Stockholm, Sweden, 2002.
- M. Prytz and A. Forsgren. Dimensioning of a multicast network that uses shortest path routing distribution trees. Technical Report TRITA-MAT-2002-OS1, Department of Mathematics, Royal Institute of Technology, Stockholm, Sweden, 2002.
- S. Ramanathan. Multicast tree generation in networks with asymmetric links. *IEEE/ACM Trans. Networking*, 4(4):558–568, 1996.
- C. No ron ha and F. Tobagi. Optimum routing of multicast streams. In *IEEE INFOCOM'94*, pages 865–873, 1994.
- G.N. Rouskas and I. Baldine. Multicast routing with end-to-end delay and delay variation constraints. In *IEEE INFOCOM'96*, pages 353–360, 1996.
- S. Sabri and B. Prasada. Video conferencing systems. *Proc. of the IEEE*, 73(4):671–688, 1985.
- H.F. Salama, D.S. Reeves, and Y. Viniotis. Shared multicast trees and the center selection problem: A survey. Technical Report TR-96/27, Dept. of Electrical and Computer Engineering, NCSU, 1996.
- H.F. Salama, D.S. Reeves, and Y. Viniotis. The delay-constrained minimum spanning tree problem. In *2nd IEEE Symposium on Computers and Communications (ISCC'97)*, pages 699–704. IEEE Computer Society, 1997a.
- 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*, 15(3):332–345, 1997b.
- C. Semeria and T. Maufer. Introduction to IP multicast routing. Internet draft (IETF), 1996.
- A. Shaikh and K.G. Shin. Destination-driven routing for low-cost multicast. *IEEE Journal of Selected Areas in Communications*, 15(3):373–381, 1997.
- R. Sriram, G. Manimaran, and C. Siva Ram Murthy. Algorithms for delay-constrained low-cost multicast tree construction. *Computer Communications*, 21(18):1693–1706, 1998.
- R. Sriram, G. Manimaran, and C. Siva Ram Murthy. A rearrangeable algorithm for the construction of delay-constrained dynamic multicast trees. *IEEE/ACM Transactions on Networking*, 7(4):514–529, 1999.
- H. Takahashi and A. Matsuyama. An approximate solution for the Steiner problem in graphs. *Mathematica Japonica*, 24(6):573–577, 1980.
- S.N. Talukdar and P.S. de Souza. Asynchronous Teams. In *Second SIAM Conf. on Linear Algebra: Signals, Systems, and Control*, San Francisco, 1990.
- T.M. Thomas II. *OSPF network design solutions*. Cisco Systems, 1998.
- D. Waitzman, C. Partridge, and S. Deering. Distance Vector Multicast Routing Protocol, RFC 1075 – IETF Network Working Group. On-line document: <http://www.ietf.org/>, 1988.
- D. Wall. *Mechanisms for Broadcast and selective broadcast*. PhD thesis, Stanford University, 1980.
- D.W. Wall. *Mechanisms for broadcast and selective broadcast*. PhD thesis, Computer Science Department, Stanford University, 1982.
- P.-J. Wan, D.-Z. Du, and Panos M. Pardalos, editors. *Multichannel Optical Networks: Theory and Practice*, volume 46 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1998.

- C.-F. Wang, C.-T. Liang, and R.-. Jan. Heuristic algorithms for packing of multiple-group multicasting. *Computers & Operations Research*, 29(7):905–924, 2002.
- B.M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.
- L. Wei and D. Estrin. The trade-offs of multicast trees and algorithms, 1994.
- S. Wi and Y. Choi. A delay-constrained distributed multicast routing algorithm. In *Proceeding of the Twelfth International Conference on Computer Communication (ICCC'95)*, pages 883–838, 1995.
- P. Winter. Steiner problem in networks: A survey. *Networks*, 17:129–167, 1987.
- P. Winter and J.M. Smith. Path-distance heuristics for the Steiner problem in undirected networks. *Algorithmica*, pages 309–327, 1992.
- W. De Zhong, Y. Onozato, and J. Kaniyil. A copy network with shared buffers for large-scale multicast ATM switching. *IEEE/ACM Transactions on Networking*, 1(2):157–165, 1993.

(C.A.S. Oliveira) SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT, OKLAHOMA STATE UNIVERSITY, 322 ENGINEERING NORTH, STILLWATER, OK 74078

E-mail address, C.A.S. Oliveira: coliv@okstate.edu

(P.M. Pardalos) DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING, UNIVERSITY OF FLORIDA, 303 WEIL HALL, GAINESVILLE, FL 32611

E-mail address, P.M. Pardalos: pardalos@ufl.edu

(M.G.C. Resende) INTERNET AND NETWORK SYSTEMS RESEARCH CENTER, AT&T LABS RESEARCH, 180 PARK AVENUE, ROOM C241, FLORHAM PARK, NJ 07932 USA.

E-mail address, M.G.C. Resende: mgcr@research.att.com