11 TELECOMMUNICATIONS NETWORK DESIGN

Anders Forsgren[1] and Mikael Prytz[2]

[1]Optimization and Systems Theory
Department of Mathematics
Royal Institute of Technology (KTH)
SE-100 44 Stockholm, Sweden
andersf@kth.se

[2]Wireless Access Networks
Ericsson Research
SE-164 80 Stockholm, Sweden
mikael.prytz@ericsson.com

Abstract: Telecommunications networks are fundamental in any telecommunications system. The network has to meet a number of criteria for the performance to be satisfactory. Hence, when designing the network, one may pose a number of optimization problems whose solutions give networks that are, in some sense, optimally designed. As the networks have become increasingly complex, the aid of optimization techniques has also become increasingly important. This is a vast area, and this chapter considers an overview of the issues that arise as well as a number of specific optimization models and problems. Often the problems may be formulated as mixed-integer linear programs. Due to problem size and problem structure, in many cases specially tailored solution techniques need to be used in order to solve, or approximately solve, the problems.

Keywords: Telecommunications optimization, network design, mixed-integer linear programming.

11.1 INTRODUCTION

Telecommunications network design is about creating a blueprint for a network. A network design is a plan for how the network should look like so that the involved parties—users, operators, regulators, etc.— will be happy with its performance and cost. Creating a network design is about choosing network structures, allocating resources, and configuring high-level parameters. It depends on the capabilities of the networking technologies, it needs a fair level of detail (but not too much since it is usually based on uncertain forecasts), and the results have to be carefully judged based on several contradicting merits.

Telecommunications networks are very complex today, in essence because they have to meet a wide range of requirements. Heterogeneous services and applications that coexist in the same network, mobility, interworking with a large number of other networks (including legacy networks), and deregulation are just a few factors that have driven complexity up. As a consequence, network design complexity has also been increased. The number of design options and possible design solutions are enormous.

The standard approach for handling the complexities of network design is to divide into subtasks and restrict possibilities, but even the subtasks can be very difficult decision problems. Many of the critical tasks relate to a practical, technical, and quantifiable problems in network design. Some examples are topology design, dimensioning (sizing/loading), configuration, routing, location of functionality, and resource allocation. Many of these problems can be naturally, and fruitfully, cast as optimization problems.

This chapter presents an overview of telecommunications network design from an optimization point-of-view, i.e., the focus is on discussing current problems in telecommunications network design where optimization models and algorithms are applicable and useful as tools. Business oriented aspects of network design, such as deciding on networking technology, and choosing targeted customers and services to offer, are not covered. Neither are technical problems that have no immediate optimization formulation (e.g., designing addressing plans).

The exposition is not complete, neither from a telecommunications network design perspective nor from an optimization application point of view. Network design in telecommunications encompasses many more aspects than what is covered here. There exists an extensive body of literature on the subject. The application of optimization to these problems also has a long history that is not described or surveyed in any detail. Nevertheless, it is our hope that this chapter can serve as a guide to the general network design issues.

11.2 BACKGROUND

To understand how optimization can be applied to telecommunications network design we begin by looking at the who, why, and how of network design: who is interested in doing a network design, why is it interesting, and what is the overall procedure.

Throughout this chapter, telecommunications network design is considered as a planning and configuration problem for a *network administrator*, which is someone, who is running, or planning to run, a telecommunications network. The network administrator can be a public operator, e.g., fixed telephony and mobile operators or Internet Service Providers (ISP). It can also be a corporation or larger company with a private network that connects its branch offices with each other and with external networks. There are also network administrators that are providers to other network administrators, e.g., access providers and leased line operators.

The planning and configuration problem is considered in the technical sense, i.e., the business-planning oriented activities of network design, such as selecting customer segments, deciding which services to offer, and determining the networking technology to use, etc., is assumed to be completed. The network is also assumed to be sufficiently large and sufficiently complex in order for there to actually be a planning problem of significance. This eliminates small home networks, which, if everything works, can be setup without planning or any detailed configuration.

Regarding network complexity there is a trend in telecommunications that strives to reduce the need for planning and configuration by developing low-cost, low-complexity, zero-configuration systems, such as ad-hoc networks. On the other hand there is an opposite trend to add new functionality and new services with increased performance requirements, all of which drives a tremendous increase in complexity of the systems. More services and applications with different inherent communication requirements, deregulation, more actors, legacy networks and technology, etc., are factors that all increase the complexity.

Costs for network components have steadily decreased. In addition, there are many unexploited potentially large markets around the world. This will give room for choices in the design, ranging from low complexity networks to smart, complex systems in optimized configuration. It is our belief that doing a good telecommunications network design will, in general, not be less difficult to do in the future than it is today.

The network design planning and configuration problem is not a static activity performed once in a predeployment phase. It is a continuous process with impacts on both the longer and shorter time scales.

There are many references on general telecommunications network design. An overview of network design issues for telecommunications networks can be found in Ericsson and Telia (1997; 1998). Long (2001) gives a similar overview directed towards Internet Protocol (IP) networks. Another general discussion covering detailed technical networking issues can be found in Stallings (1998).

The design of a network is made so as to accommodate certain traffic. The process leading to the design chosen is typically carried out in several steps. First, the network administrator business planning process yields basic demands and requirements for the network design. Examples of such requirements are served area, provided coverage, offered services, etc. We consider a long-term planning problem, where the time frame is typically many months or years up to the network's life expectancy. The life expectancy today is perhaps no more than five to ten years. The design problem is either a greenfield (design-from-scratch) problem or an expansion/modification problem in an existing network, perhaps where consideration also has to be taken to legacy equipment. The complex interaction and contradicting behavior of the design objectives discussed above, makes it very difficult to consider all aspects at the same time.

The network design process is usually separated into smaller subproblems so that each subproblem is easier to handle. An overall top-down approach is often used, beginning with a high-level design constructed to satisfy some carefully selected performance criteria. The high-level design is then refined into a detailed design, and the detailed design is then tested, for a proof-of-concept, in a lab environment (that mimics a real-life live network as closely as possible) so that all performance parameters and design objectives can be verified, see, e.g., Long (2001) or Kershenbaum (1993). It is often necessary to iterate through this process at various stages. Sometimes the high-level design has to be completely re-engineered after failing during lab verification. Many vendors now offer pre-verified and integrated solutions that minimize the effort spent in bringing the network into service.

11.3 NETWORKING

This section reviews some high-level basic technical networking concepts that are common to many different networking technologies. We describe the fundamental network components, nodes and links, and their basic properties, traffic models and demands, layered architectures and tiered network structures, protocols, and routing and forwarding. The material is brief and targeted for the optimization perspective.

A network is composed of *nodes* and *links*. The nodes represent user equipment, routers, switches, cross-connects, etc., while the links represent connections between the nodes, such as optical fibers, radio links, copper cables, etc. The distinction between nodes and links is usually apparent, but sometimes it may not be clear. In network design the nodes and links are often treated as *candidates* for where to locate nodes and links, i.e., in link topology design the nodes may exist and the full mesh of links can be candidates for where to locate links.

The purpose of the network is to shuffle *user traffic* between end users. In addition the network also transports *control traffic* for management and control of the network itself (examples are network availability and authentication information). Traffic is originated and terminated in nodes, and switched or forwarded through nodes. There may be different types or classes of traffic with different properties and requirements on the network. Traffic typically also varies over time, both over shorter and longer time scales.

11.3.1 Traffic models

To perform network capacity dimensioning it is necessary to characterize the network traffic generated by the applications and how much network resources that are needed to meet the performance requirements for the applications. We refer to such a characterization as a *traffic model.* A *traffic demand* (or just demand) is a requirement on the network design that is formulated using a traffic model. It prescribes the start- and endpoints for the traffic, often called an *origin-destination* pair for point-to-point traffic, and other parameters. The most important parameter is the *traffic volume*, i.e., how much traffic to send. It is usually given in units of some basic resource unit, or in rates of the unit (some examples are bits/s, 64 kbit/s channels, El/T1s, fractional T1s, STM-1s, wavelengths, and OC-ns). A demand can also have requirements on the maximum tolerable *delay* for end-to-end delivery of the traffic.

The transportation of traffic between nodes is carried in the links. Links may connect two nodes (point-to-point) or more than two nodes, e.g. as in an Ethernet LAN segment or when using radio. Links can be unidirectional or bidirectional, and can have many more properties. For dimensioning, in particular link sizing and loading, the *capacity* of the link is important. It is measured in bits/s or some other resource unit, and it should be related to the traffic volume of the demands.

The network is often divided into different parts that are treated separately. The division may be motivated by different functionalities, different properties, and/or different topologies in terms of network structure, geography, operator organization, etc. It may also be a way to simplify the design, to limit design effort or, more importantly, to create a network that is easy to manage. The network parts, or subnets, are often arranged in hierarchical *tiers*. The first tier, which is closest to the end user, is usually called the *access network*, while the top tier is the *backbone*. Traffic in the backbone is usually aggregated from many users, and, hence, it may benefit from trunking gains. Hence, it is important to have appropriate traffic models for each tier. A traffic demand in an access network may be described differently than a traffic demand in a backbone network. Analogously, the networking functionality is often divided into *layers* to simplify management and hide details that are not necessary for the user, see, e.g. Stallings (1998), for more details. The layers are hierarchically ordered so that a certain layer provides *service* to the layer above, while using the services of the layer below.

In a network that does not reserve resources for individual traffic sessions, a delay design objective is often more appropriate than, e.g., blocking probability. Application sessions can be described using statistical models (birth-and-death processes) that capture when traffic is sent, the rate at which traffic is sent, and for how long the application transmits, see, e.g., Bertsekas and Gallager (1992). These models are often very complicated for "bursty" applications, such as FTP (File Transfer Protocol) or web-browsing traffic, see, e.g., Paxson and Floyd (1995). Given that a network exists, it is possible, in theory, to find the distribution of queue lengths and waiting times in the links and nodes. However, due to the complicated nature of the traffic models, it is often impossible to derive analytical expressions for these distributions. Simulations are an alternative way to compute numerical estimates of delays and queue lengths.

Extending the approach to network design problems is even more complicated, since many applications actually adapt their behavior to the available resources, i.e. the applications generate *elastic traffic*. This is the case for, e.g, Transmission Control Protocol (TCP) connections in an IP network which adapt their transmission rate to the current network conditions. Some results exist on dimensioning rules for elastic traffic, see, e.g., Berger and Kogan (2000) for a single bottleneck link model.

The planning time frame for network design problems affects the necessary accuracy of traffic models. Each parameter in a traffic model has to be estimated based on forecasts of future application traffic. The increased accuracy of a more detailed model may drown in the inaccuracy of forecasting errors. This motivates the use of approximate traffic models.

One approximation in capacity network design of data networks is to ignore the coupling between traffic rates and available capacity. This approximation is accurate if a fairly tight maximum utilization constraint is enforced in the design. Another approximation is to assume a constant transmission rate, see, e.g., Bertsekas and Gallager (1992,Sec. 5.4), which at first sight may seem very inaccurate. However, in a layered network where the backbone network design is considered, there may be a large number of application sources that have been aggregated into each origin-destination pair in the backbone network. If each source contributes a small amount of traffic, then the model can be sufficiently accurate. With constant rates, a design objective can be to minimize the total delay experienced by any origin-destination pair, or it can be to minimize the cost subject to a maximum utilization constraint. For traffic with significant variations in the sending rates, models exist that compute an effective sending rate (effective bandwidth) based on estimates of means and variances, see, e.g., Kelly (1996). The effective bandwidth can then be used in capacity dimensioning as a constant rate figure.

Network traffic often takes the form of *unicast traffic*, which means point-to-point traffic. This may be considered the "typical" traffic situation, for example when two persons are having a telephone conversation. However, today's networks also have the capability of handling *multicast traffic*, which is point-to-multipoint traffic. This could be the situation when a database is replicated to a group of receivers in the network, or when a movie is distributed to a number of subscribers. The senders and receivers in a multicast session are said to belong to a *multicast group*. When taking multicast traffic in to account, one may save capacity by connecting the members of a multicast group by a tree rather than by paths. A detailed discussion of multicasting in IP networks can be found in Williamson (2000). A recent review of multicast applications and implementation challenges is given in Quinn and Almeroth (2001). Further technical details can be found in Deering (1989).

11.3.2 Traffic routing and forwarding

A fundamental problem in the design of the network is the choice of the end-point to end-point paths for the traffic demands. Such paths are constructed by the *routing* process. In a network capacity dimensioning problem it is necessary to consider how the network constructs routing paths, and how the network uses resources when forwarding traffic. It may be the case, e.g., that certain combinations of routing paths are not simultaneously realizable, which may imply that more capacity is required for the design.

Routing can be static or adaptive based on current network conditions. Note that static here means preconfigured by the network administrator - it does not, e.g., exclude the possibility to have per-hop alternative paths. In an adaptive routing network, paths may be changed from end-point to end-point. Adaptive routing works on a time scale which typically is in the order of several minutes. It may take a fairly long time before new routing information has been propagated to all nodes in the network (convergence time). The actual on-line (if any) computations of routing paths are done by routing algorithms, see, e.g., Chen and Nahrstedt (1998) for a survey of algorithms for different routing cases.

The routing is implemented by *protocols*. A protocol is a set of rules, which are implemented in nodes. A protocol entity in a node has state information about itself. It also peers with similar entities in other nodes and gather information from them. Based on state and peer information there is predefined response. An important aspect is that the protocols have a *distributed* property, in that the nodes only have access to limited information about the network. A commonly used IGP (Interior Gateway Protocol) routing protocol in IP networks is OSPF (Open Shortest Path First), see, e.g., Moy (1998a;b). This adaptive link-state protocol uses a set of pre-configured positive integer link metrics (or weights/costs). The routing paths are found as the shortest paths, computed using Dijkstra's algorithm (Dijkstra, 1959), with respect to these link metrics (avoiding failed links). The metrics are often configured to be inversely proportional to the link bandwidth capacity, see, e.g., Jones and Mitchell (2001) and the Cisco OSPF Design Guide (Cis, 2001), but this rule-of-thumb may not always yield good routing patterns, see, e.g., Fortz and Thorup (2000).

The routing gives the paths along which the traffic is sent, or *forwarded*, from one end-point to another end-point. Hence, the forwarding is done on a much shorter time scale than the routing. Forwarding can be done by two general methods. These are often referred to as circuit-, and packet-switched delivery. In the first method the delivery is grouped into sessions (or calls, flows), which are initiated through a signaling phase in which the two end-point nodes contact each other by means of an addressing scheme (telephone numbers in PSTN). After contact is established, the network *reserves resources*, if possible, for the traffic flow from end-point to end-point. These reserved resources are maintained in the network for the duration of the session (or call), which requires the network nodes to maintain state information about all active traffic sessions. Since the network resources are reserved, it is relatively easy to issue performance guarantees. However, for an application that sends traffic at a highly varying rate, the network resource utilization can be very poor.

In the packet delivery method, traffic data is divided into small units, which are given an destination address tag, and are sent directly in the network without any signaling or resource reservation. Individual nodes in the network inspect the address tag and make their own forwarding decisions. Two immediate advantages with this method are the decreased need for nodes to maintain state information, and the possibilities for increased utilization. A disadvantage is that it is more difficult to issue performance guarantees. Traffic may arrive at a node that is very busy so that it may be forced to wait in buffers or may be discarded completely.

11.4 PERFORMANCE CRITERIA

The "what is a good network design" question is difficult to answer, since it depends on many, possibly conflicting, aspects. What is good from one perspective may not be as good from another point of view. A network design engineer typically has to evaluate tradeoffs between many different design parameters. To do this successfully requires experience and a thorough understanding of the design issues and the networking technologies. Many design objectives may not be so easy to quantify or compare with other objectives. Optimization models and methods may be used as tools to support the design process, but they do not produce the full answer.

Considering the aspects above, it is nevertheless possible to identify certain major design objectives. The focus here is on objectives that are quantifiable and that can be represented in an optimization model. We discuss the most important objectives:

- performance (throughput, delay, jitter, coverage, availability);
- redundancy, resiliency, survivability;

- cost.

In addition to these, there are other factors that have to be considered. One is how manageable the network design is. How easy is it to configure and reconfigure the network? How well can fault situations be handled (detected, isolated, and repaired), and how well does the network recover in a disaster situation? Can the network be adapted to changing requirements on performance? How good is the network design in terms of security? Is the network design flexible for accounting and charging of the users? The last two points are often treated together in systems for Authentication, Authorization, and Accounting (AAA).

The ability of a network to accommodate changing and growing traffic patterns is also an objective by which a network design can be measured. A good network design should be scalable so that it does not have to be completely redesigned to accommodate growth in users and traffic volumes.

11.4.1 Performance

One of the major design objectives for a telecommunications network is its *performance*, which can be measured in many different ways. An obvious measure is the *throughput*, i.e., the aggregate traffic handling capacity in bits per second across the network. The throughput capacity should be sufficiently large to accommodate the total traffic through the network.

Another very important measure is the *application response time*, i.e., the time an application experiences in letting the network deliver traffic from one endpoint to another endpoint. Some applications, such as telephony, are very sensitive to the response time (consider, e.g., the speech quality in early implementations of satellite- or Internet-routed phone calls). Requirements on the application response time often translates into requirements on *delay*, and on *jitter* (variations in delay), that can be tolerated in the network.

Another measure of network performance is its *availability*. How frequently is an application rejected network service due to failures or congestion situations? This may again be more or less important for different applications. Requirements on availability can be translated into requirements on, e.g., the percentage of data that has to be resent or the probability that an application session is being blocked due to congestion.

A complicating issue is that requirements on response times and availability may be different for different applications existing in the same network. Networking technologies have varying support for coexisting applications with different characteristics and requirements. Traditional PSTN (Public Switched Telephony Networks) have essentially only one service class, which is very well suited to the requirements of voice traffic. ATM (Asynchronous Transfer Mode) networks offer different traffic classes (constant or varying bitrate (CBR, VBR), available bitrate (ABR), etc.) with different performance guarantees. Frame Relay networks can also handle different traffic classes. IP (Internet Protocol) networks, such as the public Internet, have some support for traffic classes and traffic prioritization, but it has not been in use on the Internet. Differentiated Services (diffserv) is now being standardized and introduced in IP networks to improve the support for different traffic classes, see, e.g., Kilkki (1999).

11.4.2 Redundancy, Resiliency, and Survivability

Network *redundancy*, *resiliency*, and *survivability* are another group of design objectives. How well does the network cope under failure situations such as router/switch failures or network link failures? This design objective can translate into requirements on reserve network resources, both in terms of standby equipment and in availability of alternate paths to route traffic on.

11.4.3 Cost

There is a tradeoff between network availability/survivability discussed above and *cost*, which is another major design objective. The total network design cost includes both equipment and capacity leasing costs, as well as the cost of support and management of the network over its lifespan. For the backbone, or the Wide Area Network (WAN), the direct capacity costs, e.g., from leased lines, can be very large and often dominates the total cost.

11.5 OPTIMIZATION OF NETWORK DESIGN PROBLEMS

The purpose of this section is to give a brief introduction to optimization models for important classes of network design problems. The problems are all formulated using mathematical optimization models. Since these involve networks and optimization they are sometimes called *network optimization* problems. This should not be confused with another common interpretation of this term in the telecommunications industry, namely the activity of tuning the performance of an existing network by carefully tweaking parameters.

We mainly focus on problems associated with the high-level network design field. Following decisions on some fundamental design issues, such as selecting networking technologies and overall architecture, the problems arise as specific, quantifiable *decision problems* having many feasible solutions. Specifically, the problems concern *capacity dimensioning of network resources* in the presence of multicast and unicast traffic, including link topology selection and link sizing/loading, and *location selection* for shared tree multicast routing core/RP nodes. The capacity dimensioning problems concern the backbone network and a physical or logical transmission layer. The traffic model for these problems is described by fixed requirements, referred to as *demands*, on bandwidth capacities between different origin-destination pairs. The problems capture the cost, availability, and performance design objectives. Different solutions can behave very differently with respect to these objectives, e.g., two designs that satisfy the performance and availability requirements may have very different total costs.

An alternative to the optimization approach is to consider several possible cases and simulate the performance of each case. This approach allows for more detailed modeling of the problem, but it is not possible to characterize the best possible solutions and there is a risk that some important design choice is missing from all cases. The optimization problems normally have to focus on some specific areas of the network design process, and the results may not be what is ultimately implemented in the network. Changes and refinements might occur during the detailed network design phase or as a result of lab verifications. This does not, however, diminish the value of finding good solutions to the mathematical decision problems. Suppose, e.g., that an optimal solution (in some sense) has been found for the mathematical decision problem. This solution then constitutes a baseline against which any detailed network design changes or modifications can be compared with. The extra cost or the added delay or the increased availability of a proposed modification can be quantified against the optimal solution, which may assist in deciding if the change should be carried through.

A general overview of network optimization models and solution methods is given in the two handbooks of Ball et al. (1995a;b). A treatment of network flow models and algorithms is given in Ahuja et al. (1993). Several telecommunications network design models and algorithms are given in Kershenbaum (1993). Extensive surveys on network design and network optimization problems, models, and solution methods can be found in Magnanti and Wong (1984), Minoux (1989), and in the recent thesis by Yuan (2001). A survey of location problems in telecommunications is given in Gourdin et al. (2001).

11.5.1 Optimization Model Components

A telecommunications network can typically be represented by a directed graph $D = (V,A)$, where V is the set of nodes and A is the set of potential arcs connecting the nodes. If we consider *unicast* traffic, there is a set of demands, K, where demand $k \in K$ may be represented by its origin node s^k and its destination node t^k are the origin and destination nodes of each demand $k \in K$. If we, for simplicity, assume that demand k requires one unit to be sent from node s^k to node t^k, the requirement on the path of the demand may be represented by a flow f^k of unit capacity from node s^k to node t^k through D. Since link capacity is almost always bidirectional, the undirected graph $G = (V,E)$ is often used to represent the network, where each edge $e = [i,j] \in E$ corresponds to two anti-parallel directed arcs $(i,j) \in A$ and $(j,i) \in A$.

A flow may be modeled by *multicommodity flow* constraints

$$Nf^k = q^k, \qquad k \in K, \tag{11.1}$$

where K is the set of demands, N is the node-arc incidence matrix of D, f^k is a vector of, continuous or binary, flow variables on A, and q^k is a vector where $q_i^k = 1$ for $i = s^k$, $q_i^k = -1$ for $i = t^k$, and $q_i^k = 0$ for $i \in V,\ i \neq s^k,\ i \neq t^k$. The choice of continuous or binary flow variables f depends on if the flows are allowed to be split or not in the network.

Although this is the "straightforward" modeling of flows through the network, the network connectivity constraints can also be represented using *cut inequalities*

$$\sum_{a \in \delta^+(S)} f_a^k \geq 1, \quad S \subset V,\ s^k \in S,\ t^k \in \bar{S}, \qquad k \in K, \tag{11.2}$$

where the *cutset* $\delta^+(S) = \{a = (i,j) \in A:\ i \in S,\ j \in \bar{S}\}$ is the set of arcs leaving S (tail in S and head in $\bar{S}$). The cut inequalities given by (11.2) are exponentially many, in general. Hence, when using this formulation, the problem is not formulated *explicitly*, but rather inequalities from (11.2) are generated as needed through the iterative solution process, see Section 11.6.

The constraints on f^k, $k \in K$, given by (11.1) or (11.2) are *separable*, in that they apply to each demand individually. In the design process, the choice of flows through the network is normally restricted by common resources, such as link capacities, which give constraints that couple the different demands. Such coupling constraints may often be modeled by binary *decision variables*. To consider a specific situation where such variables arise, assume that the network resource constraints depend on the specific performance requirement. In the capacity dimensioning problems, the constraints relate the decision variables selecting network link capacity to the decision variables connecting the network demands. Assume that multiple capacities are available for each edge, let b_e^l denote the bandwidth capacity provided by level l on edge e, and let L denote the set of capacity levels. Then we may introduce binary decision variable x_e^l, $e \in E$, $l \in L$, with the interpretation that $x_e^l = 1$ if capacity level l is chosen for edge e, and $x_e^l = 0$ otherwise. Then the capacity constraints

$$\sum_{k \in K} d^k(f_{ij}^k + f_{ji}^k) \leq \sum_{l \in L} b_e^l x_e^l, \qquad e = [i,j] \in E, \tag{11.3}$$

can be used, where d^k is the demanded bandwidth capacity for demand $k \in K$. Additional constraints may also be present, e.g., constraints that capture restrictions on routing paths.

11.5.2 *Important problem classes*

As indicated by the discussion above, a wide range of problems within network design may be formulated as as *linear mixed integer programming* (MIP) problems, i.e., an optimization problem with linear objective function and linear constraint functions, subject to integrality requirements on some of the decision variables. We outline below a few fundamental problem classes and network requirements that may be modeled within this problem class.

11.5.2.1 Minimum cost multicommodity flow. The minimum cost multicommodity flow problem is an underlying basic optimization problem. The question is how to send a set of commodities through a network at minimal cost subject to capacity constraints on the arcs. Mathematically, the problem may be formulated as

$$\begin{array}{ll} \text{minimize} & \sum_{k \in K} \sum_{[i,j] \in A} c_{ij} d^k f_{ij}^k \\ \text{subject to} & N f^k = q^k, \; f^k \geq 0, \quad k \in K, \\ & \sum_{k \in K} d^k f_{ij}^k \leq b_{ij}, \qquad [i,j] \in A, \end{array} \tag{11.4}$$

where c_{ij} denotes the cost per unit for sending through arc $[i,j]$, b_{ij} is the capacity of arc $[i,j]$, and d^k is the bandwidth required by demand k.

11.5.2.2 Uncapacitated network design—fixed charge. A fundamental network design problem referred to as uncapacitated network design with fixed charge is obtained if the capacity constraints are replaced by a cost for utilizing the arc. We then get flow variables plus one design variable per arc, i.e., a problem of the form

$$\begin{array}{lll} \text{minimize} & \sum_{k\in K}\sum_{[i,j]\in A} c_{ij}d^k f_{ij}^k + \sum_{[i,j]\in A} g_{ij}x_{ij} & \\ \text{subject to} & Nf^k = q^k,\ f^k \geq 0, & k \in K, \\ & f_{ij}^k \leq x_{ij},\ x_{ij} \in \{0,1\}, & k \in K, [i,j] \in A, \end{array} \quad (11.5)$$

where g_{ij} is the fixed cost. See, e.g., Hochbaum and Segev (1989).

11.5.2.3 Capacitated network design. The capacitated network design problem is obtained from the minimum cost multicommodity flow problem, by adding a fixed cost plus capacity constraints on the arcs. This gives

$$\begin{array}{lll} \text{minimize} & \sum_{k\in K}\sum_{[i,j]\in A} c_{ij}d^k f_{ij}^k + \sum_{[i,j]\in A} g_{ij}x_{ij} & \\ \text{subject to} & Nf^k = q^k,\ f^k \geq 0, & k \in K, \\ & \sum_{k\in K} d^k f_{ij}^k \leq b_{ij}x_{ij}, & [i,j] \in A. \end{array} \quad (11.6)$$

See, e.g., Balakrishnan et al. (1991); Barahona (1996); Bienstock et al. (1998); Holmberg and Yuan (1998); Günlük (1999); Holmberg and Yuan (2000).

11.5.2.4 Network loading problem. network loading problem

For the network loading problem, the topology of the network is given, i.e., it has been decided on beforehand which links that can be used. The question is to select capacity levels on the arcs from a give set of capacity steps. See, e.g., Mirchandani (2000).

11.5.2.5 Topology constraints. The topology of the networks is sometimes required to be of a certain type. For example, it may be of interest to use ring structures, tree structures, or some other type of particular structure. Such requirements may be included in the model, e.g., (Gavish, 1982; Holmberg and Yuan, 2000).

11.5.2.6 Routing constraints. The routing through the network may be constrained in a more complex way than through capacity levels. It may for example be required that only a restricted set of paths is used due to the fact that this has been preprocessed from a particular protocol, e.g., OSPF-routing. This may lead to more complicated problems of multilevel type. Similar complicating routing constraints may arise in networks based on restrictions on wavelengths or frequencies.

11.5.2.7 Multiperiod problems. As stated above, there is no timescale present in the problem, i.e., the demand is assumed to be static. We may consider models where the demand or capacity is forecasted for certain time intervals in the future. This may be included in the model by giving a time dimension to the design variables.

11.5.2.8 Hierarchical network design. The network design problem may be viewed on several levels. On the top level, a backbone network is designed, so as to meet different criteria. The lowest level may be a local network within a company, or even a private home. In between, there is a range of different levels of networks. By adding different levels into the model, the hierarchy of the network may be captured. See, e.g., Balakrishnan et al. (1994a;b).

11.5.2.9 Survivability. It is desirable to construct networks that are robust with respect to link or node failures. For example, if a link suddenly breaks, it is important that there are ways of rerouting the traffic so that the link failure does not lead to substantial disturbance in traffic. This may be done by selecting suitable network topology, and also by reserving spare capacity, see, e.g., Dahl (1994); Dahl and Stoer (1998).

11.6 SOLUTION STRATEGIES

The optimization models described in the previous section lead to *mixed-integer linear programs*, i.e., optimization problems on the form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \geq b, \\ & x \in Z_+^n, \end{array} \tag{11.7}$$

where Z_+^n denotes the nonnegative integral points in $\Re^n$. (Not all components of x need to have integral requirement, but to simplify the exposition, we consider the form (11.7).) We denote by S the feasible set of (11.7), i.e., $S = \{x \in Z_+^n : Ax \geq b\}$. The decision problem is to find an optimal solution x^* (we will always assume that an optimal solution exists) for a given problem instance (11.7). If all variables are binary, then there can be at most 2^n solutions, where n is the number of variables. An optimal solution can always be found by generating all 2^n candidate solutions, verify which are feasible, and pick one of the feasible solutions with the lowest objective function value. This approach is clearly only viable for very small problems.

In terms of computational complexity, the general MIP problem and the general binary programming variant are so called NP-hard problems, see, e.g., Garey and Johnson (1979). Roughly this means that there are no known efficient, i.e. polynomial time, algorithms for the problem. The network capacity design problems considered here are typically special cases of MIP problems that are still NP-hard problems (Johnson et al., 1978). Hence, in general we cannot expect to find an efficient (polynomial time) algorithm to solve practical problem instances exactly.

There are several approaches that can be used to tackle the NP-hard optimization problem (11.7). The computational difficulty of solving the problem implies that there is no known approach which is guaranteed to be efficient for all instances. Typically, it is not sufficient to consider (11.7) as such, but a successful approach has to consider special problem structure, i.e., the particular problem formulation has to be considered, e.g., (11.4), (11.5) or (11.6).

11.6.1 Approximation Algorithms and Heuristics

One approach is to use *approximation algorithms*, see, e.g., Goemans and Williamson (1997) for an application to a general class of network design problems. An approximation algorithm sacrifies optimality for efficiency, but comes with a theoretical guarantee on the worst-case percentage away that the objective value of the approximate solution is from the actual optimum.

For some classes of problems and approximation algorithms, it is possible to derive tight worst case error bounds. If the worst case error bound is tight, then the approximate solution can be considered as satisfactory. The actual optimality bound may be much tighter for a specific instance, but there is no a priori bound other than the worst case error. It is also not possible to improve on the error if the bound is not deemed as sufficiently tight.

A *heuristic algorithm* is here considered as any algorithm where no formal performance guarantee is known. Many heuristic algorithms are based on a local, or neighborhood, search strategies, such as tabu search, see, e.g, Glover (1989); Glover et al. (1993), and/or including randomization, such as simulated annealing, see, e.g., Kirkpatrick et al. (1983) or Reeves (1993). Other type of heuristics may be based on an evolutionary strategy for finding improving solutions. Genetic algorithms belong to this class of heuristics. In spite of the lack of performance guarantees, these algorithms may nevertheless produce good feasible solutions in practice although any verification of the solution quality must be provided by other means. Heuristic algorithms may successfully be applied to wide ranges of telecommunications problems. However, we will focus or discussion towards optimization methods that provide bounds on the quality of the solutions given. Heuristic methods may be a useful complement to these methods for providing good feasible solutions, i.e., to give a good upper bound of the optimal value of the problem.

11.6.2 Relaxations

A general strategy for dealing with difficult integer programming problems is to consider relaxations of the problem. A relaxed problem is typically a simplified problem which provides a lower bound on the optimal objective function value z_{MIP}. This bound can be used together with an upper bound obtained for a feasible solution to estimate the quality of the feasible solution.

The problem

$$z_{\mathrm{R}} = \begin{array}{ll} \text{minimize} & z_{\mathrm{R}}(x) \\ \text{subject to} & x \in S_{\mathrm{R}} \end{array} \tag{11.8}$$

is said to be a *relaxation* of (11.7) if $S \subseteq S_{\mathrm{R}}$ and $z_{\mathrm{R}}(x) \le c^T x$ for $x \in S$. For a general discussion of relaxations and duality for integer and mixed integer programming problems, see, e.g., Nemhauser and Wolsey (1988,Ch. II.3). The objective function value of an optimal solution, z_{R} is a lower bound on the optimal objective function value of (11.7), z_{MIP}.

The complexity of the relaxed problem (11.8) and the quality of the lower bound, depends on the choice of the objective function $z_{\mathrm{R}}(x)$ and the set S_{R}. There is a similar tradeoff here between efficiency (or effectiveness) and bound quality as there is for approximation algorithms. For a MIP problem (11.7) a natural choice is to consider the *linear programming relaxation* where $z_{\mathrm{R}}(x) = z_{\mathrm{LP}}(x) = c^T x$ and $S_{\mathrm{R}} = S_{\mathrm{LP}} = \{x \in R^n : Ax \ge b\}$. In this case the relaxed problem (11.8) becomes a linear program (LP), which can be solved efficiently. The quality of the LP lower bound may be satisfactory for some problems, but may often be poor on network design problems with discrete capacity levels, see, e.g., Prytz and Forsgren (2002).

11.6.3 Lagrangian Relaxation

Another relaxation strategy is Lagrangian relaxation, which is appropriate when the constraint structure of the MIP problem (11.7) contains a set of "nice" constraints and a set of complicating constraints such that the problem is easy (or at least less difficult) to solve if the complicating constraints are dropped. In this case the constraint set S in (11.7) is of the form

$$S = \left\{x \in Z_+^n : A^1x \geq b^1,\ A^2x \geq b^2\right\}, \tag{11.9}$$

where $A^1x \geq b^1$ are assumed to be the complicating constraints. The *Lagrangian relaxation* problem

$$\begin{array}{rll} z_{\text{LR}}(\lambda) = & \text{minimize} & c^Tx - \lambda^T(A^1x - b^1) \\ & \text{subject to} & A^2x \geq b^2, \\ & & x \in Z_+^n \end{array} \tag{11.10}$$

is a relaxation for any $\lambda \geq 0$. The relaxed problem (11.10) may be such that an efficient algorithm exists for solving it, or the relaxed problem may decompose into a set of smaller subproblems, which may lead to an overall improvement in effectiveness. The optimal objective function value $z_{\text{LR}}(\lambda)$ is a lower bound to z_{IP} for $\lambda \geq 0$. The greatest lower bound is given by the solution to the *Lagrangian dual problem*

$$\begin{array}{rll} z_{\text{LD}} = & \text{maximize} & z_{\text{LR}}(\lambda) \\ & \text{subject to} & \lambda \geq 0. \end{array} \tag{11.11}$$

The function $z_{\text{LR}}(\lambda)$ is piecewise linear and concave in λ, but it is a nondifferentiable function in general. The Lagrangian dual problem (11.11) can be solved by methods for convex, nondifferentiable optimization problems such as iterative subgradient methods, see, e.g., Hiriart-Urruty and Lemaréchal (1993). The complexity of the relaxed problem (11.10) and the quality of the lower bound may depend on which of the constraints that are relaxed. The lower bound on the optimal objective function value obtained by a Lagrangian dual problem is always at least as good as the lower bound obtained by linear programming relaxation, and the bound obtained by Lagrangian relaxation may be significantly better.

In many network design problems there are constraints relating to each demand together with a set of connecting resource constraints. Relaxing the resource constraints yields relaxed problems (11.10) that decomposes into smaller subproblems for each demand. Other relaxations are also possible, e.g., *Lagrangian decomposition* (Nemhauser and Wolsey, 1988,Ch. II.3.6).

11.6.4 Relaxation Algorithms

Relaxations of (11.7) can be used to form implicit enumeration algorithms that embody a *divide-and-conquer* strategy. For a detailed treatment, see, e.g., Nemhauser and Wolsey (1988,Ch. II.4). These algorithms are exponential in the worst-case, but together with efficient derivations of lower and upper bounds they can be applied successfully on many instances of practical size. The performance of these methods depend critically on the quality of the relaxation lower bounds and the heuristic upper bounds, as well as on the details of how the divide-and-conquer strategy is implemented.

A general iterative *relaxation algorithm* considers an initial relaxation of (11.7) with objective function $z_{\mathrm{R}}^1(x)$ and constraint set S_{R}^1, and finds an optimal solution x_{R}^1 to (11.8). If this solution satisfies $z_{\mathrm{R}}(x_{\mathrm{R}}^1) = c^T x_{\mathrm{R}}^1$ and $x_{\mathrm{R}}^1 \in S$, then x_{R}^1 is an optimal solution to (11.7). Otherwise the relaxed problem is refined by selecting a new constraint set S_{R}^2 and a new objective function $z_{\mathrm{R}}^2(x)$, such that

$$S \subseteq S_{\mathrm{R}}^2 \subseteq S_{\mathrm{R}}^1, \text{ and } z_{\mathrm{R}}^1(x) \leq z_{\mathrm{R}}^2(x) \leq c^T x, \text{ for } x \in S. \tag{11.12}$$

The refinement should be strict, i.e. either $S_{\mathrm{R}}^2 \neq S_{\mathrm{R}}^1$ or $z_{\mathrm{R}}^1(x) \neq z_{\mathrm{R}}^2(x)$ for $x \in S$. The refined relaxed problem is solved and the new optimal solution x_{R}^2 is again checked if it is optimal to (11.7) or if a new refinement has to be made.

11.6.5 Cutting Plane, Branch-and-Bound, Branch-and-Cut

A *cutting plane algorithm* is a special relaxation algorithm based on the linear programming relaxation. Here $z_{\mathrm{R}}^i(x) = c^T x$ in all iterations i and the initial constraint set is $S_{\mathrm{R}}^1 = \{x \in R^n : Ax \geq b\}$. If the optimal LP solution x_{R}^1 also satisfies integrality, i.e. $x_{\mathrm{R}}^1 \in S$, then it is an optimal solution to (11.7). Otherwise a *valid inequality* $a^1 x \geq b^1$ for S is found that separates x_{R}^1 from S, i.e. $a^1 x_{\mathrm{R}}^1 < b^1$, and S_{R}^2 is selected as $S_{\mathrm{R}}^2 = S_{\mathrm{R}}^1 \cap \{x \in R^n : a^1 x \geq b^1\}$. The problem of how to find a valid inequality that separates x_{R}^1 from S, and perhaps also finding the inequality that is most violated in some sense, is called the *separation problem*.

Another relaxation approach is based on *dividing* the problem into smaller problems that are easier to manage. Suppose S is divided into sets S^j for $j = 1, \ldots, J$ such that $S = \bigcup_{j \in J} S^j$ and let

$$z_{\mathrm{MIP}}^j = \underset{x \in R^n}{\text{minimize}} \left\{ c^T x : x \in S^j \right\}. \tag{11.13}$$

Then

$$z_{\mathrm{MIP}} = \underset{j \in J}{\text{minimize}}\; z_{\mathrm{MIP}}^j. \tag{11.14}$$

If all the sets S^j are mutually disjoint, then the division is a *partition* of S. The division of S can be done recursively by dividing or partitioning each S^j into smaller sets. This recursive division or partitioning is usually represented by a *search tree*, where the root node is the original problem and the sons, and sons of sons, etc. are the successive partitions. In a MIP problem with binary variables, a straightforward partition considers one binary variable and fixes the value of this variable to zero and one in two different subproblems, which leads to a binary search tree.

The recursive subdivisioning can be stopped for some j if either of the following three *pruning criteria* is fulfilled: (i) the subproblem is infeasible, (ii) an optimal solution to the subproblem is found, or (iii) the optimal objective function value to the

MIP problem (11.7), z_{MIP}, dominates the optimal objective function value of the subproblem (value dominance), i.e. $z_{\text{MIP}} \leq z^j_{\text{MIP}}$. Even after dividing or partitioning (11.7) the subproblems may be difficult to solve. The pruning criteria can also be difficult to apply, especially the third which assumes that z_{MIP} is known.

A natural strategy is to consider relaxations of the subproblems obtained on divisioning, which is the basis for *branch-and-bound*. The pruning criteria remain essentially the same. The second criteria can be applied if the optimal solution to the relaxed problem is an optimal solution to the subproblem, and the third criteria is modified so that z_{MIP} is replaced by $\bar{z}_{\text{MIP}}$, where $\bar{z}_{\text{MIP}}$ is an upper bound, which can be obtained, e.g., through heuristics. Branch-and-bound where relaxations are solved as subproblems can be viewed as a relaxation algorithm.

There are many aspects that affect the effectiveness of a branch-and-bound algorithm with relaxations as subproblems. In each step one subproblem node in the search tree is examined, but the overall performance of the algorithm depends on the order of the examined nodes. There may also be many ways to divide a subproblem into new subproblems, compare, e.g., the binary MIP problems where all of the remaining binary variables that have not been fixed so far are candidates for the next partition.

Branch-and-bound with linear programming relaxations is the most common branch-and-bound algorithm. It is also the basic algorithm used by most commercial MIP problem solvers. Branch-and-bound can also be used with other relaxations such as Lagrangian relaxation.

A *branch-and-cut* algorithm can be considered as a combination of branch-and-bound with linear programming relaxations and a cutting plane algorithm. Here the linear programming relaxation lower bounds are strengthened by adding valid inequalities to the problem. Note that it is possible to add inequalities that are valid only for a specific subproblem node in the branch-and-bound search tree, or that are valid for the original problem.

11.7 SUMMARY

Network design leads to many challenging optimization problems. The focus of the present chapter has been optimization problems that may be formulated as mixed-integer linear programming problems. In many cases, they are characterized by large size and cost- or constraint functions of staircase type, that may make linear programming relaxation poor. Hence, there has been significant work into specialized decomposition and relaxation methods directed towards telecommunications network design problems. The present chapter has highlighted some issues that arise within network design with respect to optimization—with an unavoidable bias towards the authors' research.

Bibliography

R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows. Theory, Algorithms and Applications*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

A. Balakrishnan, T.L. Magnanti, and P. Mirchandani. A dual-based algorithm for multi-level network design. *Management Science*, 40:567–581, 1994a.

A. Balakrishnan, T.L. Magnanti, and P. Mirchandani. Modeling and heuristic worst-case performance analysis of the two-level network design problem. *Management Science*, 40:846–867, 1994b.

A. Balakrishnan, T.L. Magnanti, A. Shulman, and R.T. Wong. Models for planning capacity expansion in local access telecommunication networks. *Annals of Operations Research*, 33:239–284, 1991.

M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors. *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*. Elsevier, 1995a.

M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors. *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*. Elsevier, 1995b.

F. Barahona. Network design using cut inequalities. *SIAM J. Optimization*, 6(3): 823–837, August 1996.

A.W. Berger and Y. Kogan. Dimensioning bandwidth for elastic traffic in high-speed data networks. *IEEE-ACM Transactions on Networking*, 8:643–654, 2000.

D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 1992.

D. Bienstock, S. Chopra, O. Günlük, and C.-Y. Tsai. Minimum cost capacity installation for multicommodity network flows. *Math. Program.*, 81(2):177–200, 1998.

S. Chen and K. Nahrstedt. An overview of quality of service routing for next-generation high-speed networks: Problems and solutions. *IEEE Network*, pages 64–79, November/December 1998.

Cisco Systems Inc., San Jose, CA, USA. *OSPF design guide*, 2001.

G. Dahl. The design of survivable directed networks. *Telecommunication Systems*, 2: 349–377, 1994.

G. Dahl and M. Stoer. A cutting plane algorithm for multicommodity survivable network design problems. *INFORMS Journal on Computing*, 10:1–11, 1998.

S. E. Deering. Host extensions for IP multicasting. Request for Comments 1112, Internet Engineering Task Force, August 1989.

E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

Ericsson and Telia, editors. *Understanding telecoummunications, vol. 1.* Studentlitteratur, Lund, Sweden, 1997.

Ericsson and Telia, editors. *Understanding telecoummunications, vol.* 2. Studentlitteratur, Lund, Sweden, 1998.

B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In *Proceedings of IEEE INFOCOM'2000*, pages 519–528. IEEE, 2000.

M.S. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, New York, 1979.

B. Gavish. Topological design of centralized computer networks - formulations and algorithms. *Networks*, 12:355–377, 1982.

F. Glover. Tabu search, part I. *ORSA Journal on Computing*, 1:190–206, 1989.

F. Glover, E. Taillard, and D. de Werra. A user's guide to tabu search. *Annals of Operations Research*, 41:3–28, 1993.

M.X. Goemans and D.P. Williamson. A primal-dual method for approximation algorithms and its application to network design problems. In D.S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company, 1997.

E. Gourdin, M. Labbé, and H. Yaman. Telecommunication and location. Technical report, Service de Mathematiques de la Gestion, Université Libre de Bruxelles, 2001.

O. Günlük. A branch-and-cut algorithm for capacitated network design problems. *Math. Program.*, 86:17–39, 1999.

J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II, Advanced Theory and Bundle Methods*. Springer-Verlag, 1993.

D.S. Hochbaum and A. Segev. Analysis of a flow problem with fixed charges. *Networks*, 19:291–312, 1989.

K. Holmberg and D. Yuan. A Lagrangean approach to network design problems. *International Transactions in Operational Research*, 5:529–539, 1998.

K. Holmberg and D. Yuan. A Lagrangean heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Research*, 48:461–481, 2000.

D.S. Johnson, J.K. Lenstra, and A.H.G. Rinnooy Kan. The complexity of the network design problem. *Networks*, 8:279–285, 1978.

M. Jones and D. Mitchell. *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols, Release 4.3*. Juniper Networks Inc., Sunnyvale, CA, USA, 2001.

F.P. Kelly. Notes on effective bandwidths. In *Stochastic Networks*, pages 141–168. Clarendon Press, Oxford, 1996.

A. Kershenbaum. *Telecommunications network design algorithms*. McGraw Hill, 1993.

K. Kilkki. *Differentiated services for the Internet*. Macmillan technical publishing, 1999.

S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

C. Long. *IP Network Design*. Osborne/McGraw-Hill, 2001.

T.L. Magnanti and R.T. Wong. Network design and transportation planning: models and algorithms. *Transportation Science*, 18:1–55, 1984.

M. Minoux. Network synthesis and optimum network design problems: models, solution methods and applications. *Networks*, 19:313–360, 1989.

P. Mirchandani. Projections of the capacitated network loading problem. *European J. Oper. Res*, 122:534–560, 2000.

J. Moy. *OSPF: Anatomy of an Internet routing protocol*. Addison Wesley, 1998a.

J. Moy. OSPF version 2. Request for Comments 2328, Internet Engineering Task Force, April 1998b.

G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1988.

V. Paxson and S. Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3:226–244, 1995.

M. Prytz and A. Forsgren. Dimensioning multicast-enabled communications networks. *Networks*, 39:216–231, 2002.

B. Quinn and K. Almeroth. IP multicast applications: Challenges and solutions. Request for Comments 3170, Internet Engineering Task Force, September 2001.

C.R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems*. Wiley, 1993.

W. Stallings. *High-Speed Networks, TCP/IP and ATM Design Principles*. Prentice Hall, 1998.

B. Williamson. *Developing IP Multicast Networks: The Definitive Guide to Designing and Deploying CISCO IP Multicast Networks*. Cisco Press, 2000.

D. Yuan. *Optimization models and methods for communication network design and routing*. PhD thesis, Division of Optimization, Department of Mathematics, Linköping University, Linköping, Sweden, 2001.