Chapter 8
Network Design with Routing Requirements

Anantaram Balakrishnan, Thomas L. Magnanti, Prakash Mirchandani, and Richard T. Wong

In Memory of Randy Magnanti

1 Introduction

The topological design and configuration of a network determines its service capabilities to transport flows of material, energy, or information effectively. These capabilities include the network's ability to route origin-to-destination flows on paths that meet performance requirements such as maximum permitted route length, time, transshipments, or likelihood of failure. To account for the interdependence between design and routing decisions, optimization models for network design jointly decide the network configuration and flow routes. However, due to economies of scale (e.g., fixed costs) in network design, optimal solutions to a basic network design model that focuses on cost minimization, without explicitly imposing routing constraints, may not meet the service requirements. For instance, when fixed costs are very high, the optimal network configuration will be sparse, implying that the routes for origin-to-destination flows on the chosen network can be long. Similarly, since facilities with higher performance capabilities (e.g., faster transport service) are more expensive, the minimum cost network design may select

A. Balakrishnan (✉)
University of Texas at Austin, Austin, TX, USA
e-mail: anantb@mail.utexas.edu

T. L. Magnanti
Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: magnanti@mit.edu

P. Mirchandani
University of Pittsburgh, Pittsburgh, PA, USA
e-mail: pmirchan@katz.pitt.edu

R. T. Wong
e-mail: r.t.wong@att.net

T. G. Crainic et al. (eds.), *Network Design with Applications to Transportation and Logistics*, https://doi.org/10.1007/978-3-030-64018-7_8

cheaper arcs that have poorer routing performance. So, for application contexts in which providing good or guaranteed end-to-end service performance is important, we must augment the basic network design model by explicitly incorporating the desired performance requirements. These requirements often take the form of constraints on the routes chosen for each origin-to-destination flow. The goals of this chapter are to explore and understand the structure of network design problems with additional route performance requirements, and discuss tailored algorithms to effectively solve the problem. For this purpose, we focus on a core model that augments the uncapacitated multicommodity, fixed-charge network design model with constraints on the arc flow variables to capture the routing requirements for each origin-destination pair. We refer to this model as the *Network Design with Routing Requirements* (*NDRR*) problem. This NP-hard problem encompasses a broad spectrum of models including the well-known Budget-constrained shortest path and Hop-constrained network design problems. We discuss modeling and theoretical issues as well as algorithmic strategies for the NDRR problem (including valid inequalities and decomposition methods), and relate these issues to prior work on special cases such as constrained shortest path and hop-constrained network design problems that can also arise as subproblems of the general NDRR problem. The simpler and special structure of these problems makes them more amenable to theoretical analysis, and have led to tailored solution techniques. We discuss opportunities to extend these results and methods to the NDRR problem and its capacitated variant. Next, we outline some practical application contexts for the NDRR model.

The NDRR model and its variants apply to transportation, telecommunication, electricity distribution, and other network-based service contexts. We briefly outline selected applications particularly in the *transportation* sector where route performance requirements can take various forms for different modes of freight or passenger transport.

- The *vehicle routing* problem with delivery deadlines (e.g., Desaulniers et al. 2014; Vidal et al. 2013) requires finding minimum-distance vehicle routes to deliver products from depots to geographically dispersed customers, with each customer requiring delivery by a specified time. We can view this problem as a NDRR problem (with additional constraints to ensure that the routes are cyclic) in which each required delivery corresponds to a commodity to be dispatched from a depot to a customer location; selecting an arc from i to j corresponds to routing a truck (carrying orders for multiple customers) between these two locations at a fixed cost equal to the distance from i to j. The delivery deadline for each customer imposes an upper limit on the time that the truck assigned to this customer takes to reach the customer location after departing from the depot, including the time for intermediate deliveries. Thus, the route performance requirement for each customer is the total time for the depot-to-customer route.
- *Package* and *less-than-truckload* carriers need to decide when to dispatch trailers (loaded or empty) between various hub or transshipment locations, and how to route shipments on the chosen services (e.g., Malandraki et al. 2001; Estrada

and Robuse 2009). Chapter 12 of this volume on Service Network Design and Chap. 14 on Motor Carrier Network Design elaborate on the optimization models that arise in this context. Moving trailers between any pair of hubs incurs fixed costs that depend both on distance and the frequency of these services. More frequent movements increase cost, but reduce the waiting (at hub) and/or total travel time for shipments. Shipments may have different priorities, with some requiring time-definite deliveries and others having less stringent requirements. This service design problem can be viewed as a NDRR problem defined on a time-space network whose arcs represent trailer movements and connections (including waiting) at hubs, with the additional requirement that shipments must be routed within their guaranteed origin-to-destination transit times.

- *Airline crew scheduling* (e.g., Gopalakrishnan and Johnson 2005) entails deciding the pairing or duty cycle for each crew member to ensure that each scheduled flight has the required complement of crew members while satisfying crew work rules. The cost of each duty cycle depends on the crew member's assigned flight legs, deadheads, and layovers at intermediate locations. Federal regulations and union rules limit the total duration and possibly the number of layovers in each duty cycle. In the NDRR framework, the commodities are crew members, and the routes correspond to deciding the sequence of flight legs for each duty cycle such that the cycle duration (and number of layovers) does not exceed the permitted value.
- *Service design* for *freight railroads* (e.g., Zhu et al. 2014; see also Chap. 13) requires, as one of its components, designing an effective blocking network. This problem entails selecting a limited number of blocks (sequences or paths of train-service legs, which can be viewed as logical links between yards, on which groups of railcars travel together) for routing shipments (e.g., Barnhart et al. 2000; Ahuja et al. 2007). To limit the number of times railcars are reclassified, i.e., moved from one block to the next block at an intermediate yard, during their origin-to-destination trip, the number of arcs in each shipment's trip plan must not exceed a pre-specified upper limit (that can vary by shipment).
- *Liner container shipping* companies must decide the cyclic routes for their ships, the frequency of service on each route, and the movement of containers between origins and destinations on the chosen services (e.g., Agarwal and Ergun 2008; see also Chap. 15). The route for each container consists of a sequence of sailing legs on different services, with transshipment from one service to the other at intermediate ports. Transshipments are expensive due to cargo damage or loss, handling, and storage; they also increase the origin-to-destination transit time because containers have to wait at the intermediate port for the next scheduled service. So, shipping companies seek to design their service network so that they can transport cargo subject to restrictions on the transit time and number of transshipments (e.g., Balakrishnan and Karsten 2017; Karsten et al. 2017).

Analogous applications with route performance requirements are also pervasive in *telecommunications* network planning. The performance specifications, often referred to as *Quality of Service* (QoS) requirements, stem from the need to

limit the 'latency' or end-to-end transmission delay for the many vital and time-sensitive traffic flows on telecommunication networks, including voice-over-IP, distributed game playing, transmission of financial information, and emergency communications. The latency depends on the speeds of the links on the path as well as the number and speeds of intermediate routers/switches. Moreover, we must route critical communications over 'reliable' paths having low probability of link failures or packet switching loss. For certain applications such as multicast broadcast networks, QoS considerations impose additional requirements such as limiting the number of links or hops on the paths from the root node (typically, the message source) to every other node (distribution points or destinations).

The problem of deciding the optimal network configuration (or upgrading an existing network) while ensuring adequate routing performance also arises in contexts such as energy distribution (e.g., De Boeck and Fortz 2017) and supply chain networks, and applies to various scheduling problems that we can define over virtual (versus physical) networks. For instance, we can view the parallel machine, non-preemptive scheduling problem where jobs have different release times and deadlines and require sequence-dependent change-over times (or costs) as the problem of identifying the least cost star network (with as many branches as the number of machines) that spans all the job nodes, with restrictions on the maximum time to reach each job node from the root node. Other applications of the NDRR framework include managing feature addition during a product's life-cycle (e.g., Wilhelm et al. 2003) and optimal path configuration for radar avoidance (e.g., Zabarankin et al. 2001).

Given such widespread applications of the NDRR problem, we focus on how to effectively model and solve this problem. Section 2 provides a classification of network design problems with routing requirements, and formulates the core version that we will address in the remainder of the chapter. We also discuss the challenges in solving the problem, and outline some related threads of theoretical research on the problem's difficulty. Section 3 outlines a polyhedral approach for effectively solving the general network design problem with routing requirements that combines problem reduction, model strengthening, and cutting planes. Section 4 addresses two notable special cases of the problem—constrained shortest path and hop-constrained network design problems—that can arise as subproblems of the general problem. We also describe illustrative tailored solution approaches that exploit the special structure of these problems. Section 5 discusses decomposition methods to solve the general problem, including Lagrangian relaxation, column generation, and Benders decomposition. We also discuss how the preceding methods can be extended to the capacitated variant of the problem that imposes arc capacity constraints in addition to routing requirements. Section 6 provides Bibliographical Notes on prior literature related to the discussions in the following sections. Section 7 concludes the paper with a summary of key observations and learnings about the NDRR problem, and some thoughts on future research directions.

2 Problem Classification and Model Formulation

Network design encompasses a vast array of models that differ in their features and assumptions depending on the application context. Section 2.1 briefly outlines a framework to classify network design problems based on their structure and assumptions. Section 2.2 elaborates on the types of additional flow constraints (besides demand, supply, and flow conservation constraints) that routing requirements may impose. Section 2.3 presents the integer programming formulation for the NDRR problem that we study, and Sect. 2.4 provides insight into why the problem is challenging and why even some of its simpler special cases are difficult.

2.1 Model Classification

The two core decisions for network design are: (1) which arcs, from the given set of candidate arcs, to include in the design, and (2) how to route the origin-to-destination flows on the chosen arcs so as to satisfy demand. We refer to the corresponding decision variables as *design variables* and *flow* or *routing variables*. Two types of constraints are common to all network design models: flow conservation constraints on the flow variables (including demand and supply constraints), and forcing constraints to relate the design and routing decisions, i.e., to ensure that flow is only routed on arcs that are included in the design. We can differentiate network design problems along the following four main dimensions, based on the arc and flow characteristics and requirements.

- *Directed* arcs that can carry flows only in the arc's direction versus *Undirected* edges that permit flows in both directions. Generally, network design models (without any additional valid inequalities) over undirected networks tend to have weaker LP lower bounds than those for directed networks (see, for example, Balakrishnan et al. 1989).
- *Multiple commodities*, distinguished by their origins and destinations, costs, and other characteristics, versus a *Single* homogeneous commodity that can be supplied by any source to a destination. With multiple commodities, the origin-destination demand pattern can be arbitrary or have special structure (e.g., single source, single destination, or complete demand between every pair of nodes). For network design, multicommodity formulations can be tighter (e.g., Rardin and Choe 1979; Vanderbeck and Wolsey 2010).
- *Non-bifurcated* flows that must be routed on a single path from each commodity's origin to destination versus *Bifurcated* flows that permit splitting the required flows among multiple origin-to-destination paths. Ensuring that flows are non-bifurcated requires defining binary variables to define the path for each commodity, making the problems more difficult to solve.
- *Additional constraints*: Network design applications in practice may impose additional constraints besides the flow conservation and forcing constraints of

the basic uncapacitated model. These constraints fall into two main categories—configuration constraints and flow restrictions. Configuration or design constraints involve only the design variables, and define the permissible configurations. For instance, some applications require the design to be a tree network (e.g., for multicasting) while others seek a network that is the union of cycles (e.g., container ship routes, fiber optic ring networks). Flow or routing restrictions limit the routing options by imposing constraints on the flow variables. We will discuss these latter constraints in more detail in Sect. 2.2.

Within this framework, the model can accommodate several other variants such as different objective functions (e.g., maximizing profits with the flexibility to selectively meet demands, or minimizing the number of transshipments) and incorporating node attributes (costs, waiting or processing times, other capabilities). In this chapter, we focus on directed, multicommodity problems with non-bifurcated flows together with additional constraints that we discuss next to account for service requirements.

2.2 Routing Requirements

We capture routing and service requirements by constraining the routes on which commodities can flow. We can broadly classify such constraints as inter-commodity constraints or intra-commodity constraints. *Inter-commodity* constraints enforce joint requirements on the flows of multiple commodities. The most common example is the arc capacity constraint to ensure that the total flow of all commodities on an arc does not exceed the arc's capacity. Other examples include situations in which using an arc for one (or more) commodity on an arc necessitates either not routing another commodity (or subset of commodities) or co-routing another commodity on that arc. For instance, in the rail freight industry, policy and technological restrictions prohibit transporting certain combinations of commodities on the same arc. Likewise, in crew scheduling, some organizations favor keeping crew members from different occupations (commodities) together as a team for multiple trips. *Intra-commodity* constraints refer to flow constraints that involve flow variables from a single commodity. Many of the applications discussed in Sect. 1 fall into this category since they impose performance or service requirements on each origin-to-destination flow, which in our model corresponds to an individual commodity. These applications vary in the performance *metric* they use to ensure that the solution meets service requirements. Further, the metric is often additive, i.e., the total value of the metric for a path, which the constraint seeks to limit, is the sum of the metrics of the arcs and nodes on this path. We next list some common metrics.

- *Time*: In transportation applications, the metric for each arc (node) is often the transportation (transshipment) time. Upper bounds on the transit time from origin to destination, which is the sum of traversal times of the arcs and nodes on the

route, can stem from delivery deadlines, product perishability, or the need to reduce in-transit inventory.

- *Distance*: Each arc has an associated distance, and operational or service considerations may require selecting origin-to-destination routes whose distance does not exceed a pre-specified value that can vary with the origin-destination pair.
- *Cost*: Arcs and nodes may have associated costs or other financial metrics (different from those in the cost minimization objective function) for using the arcs or for processing at the nodes. Associated constraints, sometimes called *budget* constraints, impose upper limits on the total cost for each origin-to-destination route.
- *Transshipments*: Many contexts require limiting the number of intermediate transshipments on origin-to-destination routes, for instance, to avoid excessive handling and to regulate the effort and time for processing at nodes. By associating a metric whose value is one for each arc, the total value for any origin-to-destination route is the number of *hops*, i.e., number of arcs on this route; the service requirement imposes an upper bound on this value.
- *Reliability*: In contexts where arcs (or nodes) can fail, a natural service requirement is that the reliability of any chosen origin-to-destination route, defined as the likelihood that this route is operational, must exceed a pre-specified threshold value. Defining the performance metric of each arc (node) as the logarithm of the probability that the arc (node) will be operational and assuming that arc (and node) failures are independent, the service requirement imposes a lower bound on the sum of the arc (node) metrics on any route.

Additional constraints on flow variables may also arise due to operational restrictions or policies that govern routing decisions. For instance, we can add constraints to model logical conditions such as the following: if the route contains arcs from a specified subset, it must not include any arc (or must necessarily include every arc) from another subset. These constraints arise in transportation contexts (e.g., shipment routing for different materials on railroads) and also to impose special configuration requirements such as requiring multiple possible routes for each commodity (e.g., Grotschel et al. 1995; Balakrishnan et al. 2009).

In each of the above examples, the requirement (e.g., maximum transit time, maximum permitted number of hops, minimum required reliability) can vary by origin-destination pair. Further, if we classify traffic flows between an origin and destination into multiple types based on their priorities or service characteristics, we can define separate commodities for each flow type, permitting finer-grained differentiation of routing requirements. Finally, we can readily incorporate performance metrics associated with nodes by simply adding the value of each metric corresponding to a node to the metric of each arc that is incident to (or from) that node. Next, we present an integer programming formulation for the network design problem with routing constraints, and discuss some of its special cases.

2.3 Model Formulation

As noted in Sects. 2.1 and 2.2, network design problems with routing restrictions have many different variants. Our main focus in this chapter is to understand the effects on problem structure and solution strategies when we impose performance or service requirements on origin-to-destination routes in network design solutions. Accordingly, we consider the core multicommodity, fixed charge, uncapacitated network design problem, requiring non-bifurcated flows for each commodity, augmented with intra-commodity routing constraints.

We use the following notation to formulate this optimization problem. Let $\mathscr{G} = (\mathscr{N}, \mathscr{A})$ be the directed graph on which the problem is defined. Node set $\mathscr{N}$ consists of n $(= |\mathscr{N}|)$ nodes representing origin, destination, or transshipment nodes, and arc set $\mathscr{A}$ contains arcs that are available for installation and use. Let $\mathscr{K}$ denote the set of commodities. Commodity $k \in \mathscr{K}$ originates at node $O(k)$ and terminates at node $D(k)$. Commodities may be further distinguished by their routing and service requirements, i.e., we can have multiple commodities with the same origin and destination but different routing constraints. Without loss of generality (since the network is uncapacitated), we scale each commodity's demand to one but permit the routing or flow cost to vary by commodity. Define $\mathscr{N}_i^+ = \{j \in \mathscr{N} : (i, j) \in \mathscr{A}\}$ and $\mathscr{N}_i^- = \{j \in \mathscr{N} : (j, i) \in \mathscr{A}\}$ as the subsets of downstream and upstream neighbors for each node i.

The network design problem has two sets of binary decision variables: (1) *design* variable y_{ij}, for each arc $(i, j) \in \mathscr{A}$, that takes the value one if the solution includes arc (i, j) in the design, and is zero otherwise; and, (2) *routing* or flow variable x_{ij}^k, for arc $(i, j) \in \mathscr{A}$ and commodity $k \in \mathscr{K}$ that equals one if the solution routes commodity k on arc (i, j). Let f_{ij} and c_{ij}^k respectively denote the non-negative fixed cost for using arc (i, j) and flow cost for routing commodity k on this arc. We permit imposing m^k different routing constraints for each commodity k, one corresponding to each performance metric of interest (as discussed in Sect. 2.2). For each metric $m = 1, 2, \ldots, m^k$, let q_{ij}^{km} denote the non-negative coefficient or *weight* of the routing variable x_{ij}^k in the m^{th} constraint, and let Q^{km} be the *weight limit*.

Using this notation, we can formulate the *Network Design problem with Routing Requirements* (NDRR) as the following integer program, denoted as model [*NDRR*].

$$\text{Minimize} \quad \sum_{(i,j)\in\mathscr{A}} (f_{ij} y_{ij} + \sum_{k\in\mathscr{K}} c_{ij}^k x_{ij}^k) \tag{8.1}$$

subject to:

$$\sum_{j\in\mathscr{N}_i^+} x_{ij}^k - \sum_{j\in\mathscr{N}_i^-} x_{ji}^k = \begin{cases} 1 & \text{if } i = O(k), \\ -1 & \text{if } i = D(k), \\ 0 & \text{otherwise,} \end{cases} \quad \forall\, i \in \mathscr{N}, \tag{8.2}$$

$$x_{ij}^k \le y_{ij}, \qquad \forall\, (i, j) \in \mathscr{A}, k \in \mathscr{K}, \tag{8.3}$$

$$\sum_{(i,j)\in\mathscr{A}} q_{ij}^{km} x_{ij}^{k} \le Q^{km}, \qquad \forall k \in \mathscr{K}, m = 1, 2, \ldots, m^k, \tag{8.4}$$

$$x_{ij}^{k} = 0 \text{ or } 1,\ y_{ij} = 0 \text{ or } 1, \qquad \forall (i, j) \in \mathscr{A}, k \in \mathscr{K}. \tag{8.5}$$

The objective function (8.1) minimizes the total fixed and routing costs. Constraints (8.2) impose *flow conservation* at every node for each commodity. Together with the *integrality* constraints (8.5) on the routing variables x_{ij}^k, the flow conservation equations ensure that the flow solution selects a single origin-to-destination route for each commodity k. The *forcing* constraints (8.3) relate the design and routing decisions; they specify that we can route commodity k on an arc (i, j) only if the design includes this arc (after incurring its fixed cost). Although it is possible to represent this condition using fewer 'aggregate' forcing constraints of the form $\sum_{k\in\mathscr{K}} x_{ij}^k \le |\mathscr{K}| y_{ij}$, one for each arc, the disaggregate version (8.3) yields a tighter linear programming (LP) relaxation (see Balakrishnan et al. 1989). The *routing* constraints (8.4) require the total weight of commodity k's route to be less than or equal to the weight limit, for each metric $m = 1, 2, \ldots, m^k$. Finally, constraints (8.5) require the design and flow variables to be binary. Note that we have assumed, for notational simplicity, that every commodity can flow on each arc in the set $\mathscr{A}$. If a commodity is prohibited from flowing on certain arcs (e.g., due to operational or technological issues), we can eliminate the corresponding flow variables from the formulation.

The [*NDRR*] model has two interesting special cases that we will study further Sect. 4. First, if the set $\mathscr{K}$ contains just one commodity, then the problem reduces to finding the shortest path that satisfies all the routing constraints for this commodity. This special case, called the *Constrained Shortest Path* (CSP) problem, is interesting both because it has direct applications in a variety of practical settings and because it often arises as a subproblem in decomposition algorithms for the NDRR problem and other models. Section 4.1 discusses its properties and solution algorithms. In the second interesting and relevant special class of problems, which we call *Hop-constrained* problems, there is only one weight metric, with weight $q_{ij}^{k1} = 1$ for every commodity k and arc (i, j), and the weight limit for commodity k is the maximum allowable number of arcs (or hops) on the commodity's origin-to-destination route. We refer to this limit as the *hop limit*. Among such hop-constrained problems, we consider path and tree versions. For the latter version, we are given a root node and seek a minimum spanning tree such that none of the nodes are more than a specified number of hops away from the root node in the chosen tree. This problem, which we call the *Hop-constrained Minimum Spanning Tree* (*HCMST*) problem, can be modeled as a special case of the NDRR problem in which $|\mathscr{K}| = n - 1$, the root node is the common origin node for all commodities, and every other node is a destination. The Hop-constrained Steiner Tree problem generalizes the HCMST problem by requiring only a subset of nodes to be connected to the root node, i.e., commodities are defined only for a subset of non-root nodes. As we noted in Sect. 2.2, hop constraints are common in many practical applications. Section 4.2 discusses properties and solution methods for hop-constrained problems.

We conclude this discussion by noting that model [*NDRR*] is an arc-flow formulation of the NDRR problem in which the flow variables model each commodity's route as a sequence of arcs. As an alternative, we might consider a path-flow formulation in which the (binary) flow variables represent the choice of origin-to-destination path for each commodity. By limiting (a priori) the available paths to those that are feasible, i.e., satisfy all the routing constraints, the model only requires path selection and forcing constraints. The path-flow formulation has the advantage of having a tighter LP relaxation, and hence higher LP lower bounds, than the arc-flow model. However, it also has the significant drawback of requiring an exponential (in the size of the network) number of path-flow variables. One approach for overcoming this drawback is to use column generation to solve the problem; this approach iteratively generates promising paths based on the dual values for the current solution. However, the subproblem to generate columns is a CSP problem, which is itself NP-hard. In Sect. 6, we discuss the column generation approach to solve NDRR problems.

2.4 Challenges in Solving the NDRR Problem

The NDRR problem is challenging to solve (compared to the basic uncapacitated fixed-charge network design problem) due to the added routing restrictions which complicate the problem structure and make it difficult to even find feasible solutions. We next discuss these issues.

Problem Complexity Adding routing requirements to even simple problems can make them computationally difficult and intractable. For instance, the Shortest Path problem can be efficiently solved, but if we add just one routing constraint, the resulting problem, often called the Budget-constrained Shortest Path (BCSP) problem, is NP-hard (see Garey and Johnson 2002). Similarly, although the Minimum Spanning Tree problem is polynomially solvable, if we add hop constraints (to limit the number of arcs on the path from a root node to every other node), we obtain the HCMST problem, which is NP-hard even if the number of hops is limited to two. The result follows using a transformation from the uncapacitated facility location problem (Dahl 1998). Since the BCSP and the HCMST problems are both special cases of network design with routing requirements, the NDRR problem is also NP-hard.

Multiple Routing Requirements If a commodity has more than one routing requirement, then even finding a feasible solution is NP-hard (Balakrishnan et al. 2020, Grandoni et al. 2014). The four-node example shown in Fig. 8.1, with two routing requirements for a commodity, illustrates this issue. In this example, the commodity originates at node 1 and terminates at node 4. The numbers next to each arc show the arc's cost and its two weights, one for each metric, in the two routing constraints. The weight limits for both metrics is five. The minimum cost path from node 1 to node 4 is 1-2-4, but this path does not satisfy either routing

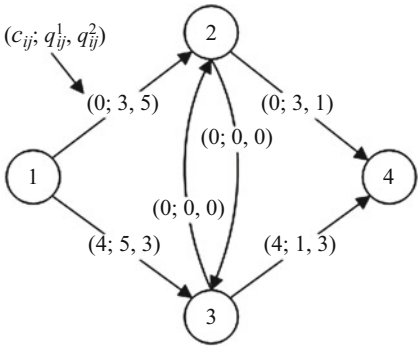

Fig. 8.1 Example with multiple routing requirements

constraint. If the problem contains only one routing constraint, say, only the first constraint, we can readily verify if the problem instance is feasible by finding the shortest "weight" path, using the arc weight for the first metric as the length of each arc. In this example, the shortest weight path for the first metric is 1-2-3-4, with a total weight of 4. Hence, the problem instance is feasible with just the first routing constraint. However, this path does not meet the second routing requirement. Similarly, the shortest weight path 1-3-2-4 using the second set of weights as arc lengths satisfies the second routing requirement, but not the first. The paths 1-2-4 and 1-3-4 satisfy neither routing constraint. So, this problem instance is infeasible if we impose both routing constraints. Observe that we had to examine every origin-to-destination path to determine that the instance is infeasible, suggesting that, with multiple routing requirements, even verifying feasibility is difficult.

Worst-Case Integrality Gap As another indicator of the added problem difficulty when we incorporate a routing constraint, consider again the BCSP problem. We know that the network flow formulation of the unconstrained Shortest Path problem has integer extreme points; so, the LP relaxation of this problem has an integer optimal solution with zero integrality gap. However, when we add a budget constraint, the integrality property no longer holds. That is, the optimal solution to the LP relaxation of the BCSP can be fractional, necessitating the explicit addition of integrality constraints. The fractional LP solution arises because the solution can satisfy the budget constraint 'on average' by routing partial flows on two different paths—one with low cost but high weight and another with higher cost but lower weight. With more than one weight constraint, the LP solution can be a convex combination of more than two paths, none of which satisfy all the weight constraints. These observations suggest that, for general fixed-charge network design problems, the gap between the optimal IP and LP values may be higher when we include routing requirements compared to the gap without these restrictions.

Finally, note that for the CSP problem (and the more general NDRR problem), if the problem imposes only one routing requirement for each commodity (as in the above example), then the original problem is feasible if and only if the LP relaxation is feasible. Moreover, we can construct a feasible solution from the LP solution by routing each commodity over its feasible route on which the LP solution routes a fractional flow, and setting the design variables on all the arcs belonging to these routes to one. However, these observations do not hold when there are two or more routing requirements for a commodity. In this case, the LP relaxation may be feasible even if the original integer program is not. So, we cannot readily construct a feasible IP solution from the LP solution.

3 Solving the NDRR Problem

As our discussions in the previous sections indicate, adding routing requirements to fixed-charge network design models makes them more difficult to solve. Therefore, effectively solving these problems requires exploiting the embedded structure of these problems to reduce problem size, raise lower bounds, and develop specialized solution methods to accelerate performance. We focus on methods that can either solve the problem optimally or provide guarantees of solution quality. We do not consider the many possible meta-heuristic approaches that solve the NDRR problem approximately but do not assure near-optimality. This section describes a cutting plane approach to solve the arc flow formulation of the general NDRR problem based on the recent paper by Balakrishnan et al. (2017), henceforth abbreviated as BLM, which is the only paper to date that addresses the general version of this problem. In Sect. 3.1, we discuss problem reduction methods to eliminate decision variables and also tighten the NDRR problem formulation. Section 3.2 discusses valid inequalities and outlines polyhedral results that underlie a cutting plane approach for the NDRR problem. Computational tests using this method, together with problem reduction and optimization-based heuristics, demonstrated that this approach can significantly reduce computational time compared to applying standard solution procedures.

3.1 Problem Reduction

Problem reduction refers to methods that we can apply a priori (before solving the problem) to fix the values for some decision variables based on feasibility requirements or properties of optimal solutions. For the NDRR problem, these techniques entail either eliminating (i.e., fixing at zero) some commodity routing or design variables, or requiring a commodity to flow through an arc (i.e., fixing the corresponding commodity flow and design variables to one). Such restrictions not only reduce the size of the model formulation (by eliminating variables and

constraints), but can also tighten the model, i.e., increase the optimal value of the LP relaxation. We next discuss some intuitive approaches to reduce the NDRR problem. For this discussion, for any commodity $k \in K$ and metric $m = 1, 2, \ldots, m^k$, let L^{km}_{ab} denote the total length of a shortest weight path from node a to node b in the given graph, using the arc weight q^{km}_{ij} as the length of each arc (i, j).

Eliminating Flows To determine if we can eliminate the flow of commodity k on an arc (i, j), we check if, for each metric m, the shortest weight path from $O(k)$ to $D(k)$ containing this arc, consisting of the shortest weight path from $O(k)$ to node i, arc (i, j), and the shortest weight path from node j to $D(k)$ has total weight not exceeding the weight limit Q^{km}. If not, i.e., if:

$$L^{km}_{O(k)i} + q^{km}_{ij} + L^{km}_{jD(k)} > Q^{km} \tag{8.6}$$

for at least one metric $m = 1, 2, \ldots, m^k$, then arc (i, j) does not belong to any feasible origin-to-destination path for commodity k, and so we can eliminate variable x^k_{ij} and the associated forcing constraint (8.3) from the model formulation. Further, if the above test eliminates the flow of commodity k on all arcs incident to (or from) a node i, we can eliminate the flow conservation constraint in (8.2) for this commodity at node i. Finally, if we find that only one commodity k' can flow on an arc (i, j), then we can omit the design variable y_{ij} from the formulation, simply add the fixed cost f_{ij} to the routing cost $c^{k'}_{ij}$ for this commodity, and eliminate the corresponding forcing constraint in (8.3).

Fixing Flows For any arc (i, j) and commodity k, let $L^{km}_{ab}(\mathscr{A}\backslash(i, j))$ denote the total weight (or length) of a shortest weight path from node a to node b after deleting arc (i, j) from the given graph, and using the arc weights for metric m as arc lengths. Then, if

$$L^{km}_{O(k)D(k)}(\mathscr{A}\backslash(i, j)) > Q^{km} \tag{8.7}$$

for at least one metric $m = 1, 2, \ldots, m^k$, i.e., the shortest weight path that does not include arc (i, j) is not feasible for some metric, then commodity k must necessarily flow on arc (i, j). In this case, we can set $x^k_{ij} = y_{ij} = 1$ in the formulation, drop these two variables, and omit all the forcing constraints (8.3) for arc (i, j). Further, since commodity k can only be routed on an elementary path (without cycles) in any optimal solution, we impose the restriction that this commodity cannot flow on any other arc that is incident from node i or to node j. So, we can drop the variables $x^k_{i'j'}$ for all arcs (i', j') with either $i' = i$ or $j' = j$ but not both, and eliminate the forcing constraints for these variables.

Fixing Routes For any commodity k and any feasible (satisfying routing requirements) origin-to-destination path p for this commodity, let $C(p) = \sum_{(i,j)\in p} c^k_{ij}$ and $TC(p) = \sum_{(i,j)\in p}(f_{ij} + c^k_{ij})$ respectively denote the routing cost and the *total* (fixed plus routing) cost of this path. The total cost essentially assigns the fixed

cost of every arc on path p to commodity k. Suppose the problem instance has a unique feasible $O(k)$-to-$D(k)$ path p^* whose total cost $TC(p^*)$ does not exceed the routing $C(p)$ for any other feasible path p for commodity k. Then, the NDRR problem has an optimal solution that routes commodity k on path p^*, implying that we can fix the route for commodity k. To apply this test, we use a CSP algorithm, with commodity k's routing requirements as constraints, to identify path p^* and the other paths. Specifically, p^* is the constrained shortest path using the total cost $(f_{ij} + c_{ij}^k)$ as the length of each arc (i, j). To find the other paths p, we apply the following procedure. For every arc (i, j) in p^*, we delete (i, j) from the network, find the constrained shortest path using routing costs as arc lengths, and choose p' as the least (routing) cost path among these paths. If $TC(p^*) \leq C(p')$, then we can eliminate commodity k (and all its associated variables and constraints) from the problem formulation since this commodity must flow on path p^*; in this case, we fix to one the values of the design variables y_{ij} for all arcs (i, j) on p^*, and omit the forcing constraints for the underlying flow variables.

The above discussion illustrates the two broad approaches to reduce the problem size by eliminating or fixing variables—based on weight feasibility, such as the first two conditions for eliminating or fixing flows, or on cost (optimality) such as the third test for fixing routes. These approaches can greatly improve solution performance for the overall NDRR problem both by reducing model size (variables and constraints) and by raising the LP lower bounds. The effectiveness of these tests depends on the characteristics of the problem instance. The feasibility tests (for eliminating or fixing flows) are likely to be more effective if the weight limits in the routing requirements are somewhat stringent (tight), the network is sparse, and the arcs vary widely in their weights. For instance, for the hop-constrained network design problem (that requires each commodity to flow on a route that uses no more than a prespecified number of arcs or hops), if the hop limits are small and the network is sparse, the test to eliminate flows can be very effective since many arcs may not belong to any low-hop path from the commodity's origin to destination. If, in addition to the hop constraint, the route is also subject to a more general weight constraint and the distribution of arc weights has wide dispersion, then the problem instance may permit further reduction. We conclude this discussion by noting that the first two methods, for eliminating and fixing flow variables, also apply to the CSP problem.

3.2 Valid Inequalities and Composite Algorithm for the NDRR Problem

Polyhedral approaches have proven to be very effective to solve several notoriously difficult integer programming problems, including many variants of network design. The success of these methods rests on using strong problem formulations (i.e., formulations with small gaps between the optimal value of the integer program and

its LP relaxation) and on developing tight valid inequalities, preferably facets, that can be added to cutoff fractional solutions. Often, tailored valid inequalities, that exploit insights about the polyhedral structure of the underlying problems, are most effective and yield significant improvements in the LP lower bounds. Previously, researchers have studied the polyhedral structure and developed tighter formulations for certain special cases of the NDRR problem such as hop-constrained network design (see Sect. 4). For the general NDRR problem, we next summarize the work of Balakrishnan et al. (2017) (*BLM*) who developed and successfully applied several classes of valid inequalities to solve the problem's arc flow formulation [*NDRR*].

The LP relaxation of model [*NDRR*] may achieve a much smaller optimal value than the integer program by splitting the flow of a commodity across multiple origin-to-destination paths. This flow splitting occurs for two reasons: (1) by splitting flows, the LP relaxation can select fractional values for the design (y_{ij}) variables, thus only partially absorbing the arc fixed costs; and (2) the LP solution can reduce the routing cost component of its objective value by partially routing a commodity's flow on paths that have low routing costs but high weights since it is only required to meet the route restrictions 'on average' (as illustrated in Sect. 2.3). The first reason applies more generally to other fixed-charge problems, whereas the second reason is specific to the NDRR problem. BLM propose three broad families of inequalities, called *Route Composition*, *Contingent Routing*, and *Multicommodity Design* inequalities, to reduce flow splitting, and prove that some specific versions of these inequalities are facet-defining. The first two inequality classes focus on reducing (partial) flows on infeasible paths, thus addressing the second reason above, whereas the last class addresses both reasons. To motivate these inequalities, provide intuition, and highlight the underlying principles, we discuss a few illustrative versions of these inequalities; BLM provide a more comprehensive treatment.

Route Composition inequalities These commodity-specific inequalities impose the condition that the route for a given commodity must not contain more than a certain number of arcs from a carefully chosen subset of arcs. Since each routing requirement resembles the capacity constraint in a knapsack problem, constraints analogous to the knapsack cover inequality (e.g., Nemhauser and Wolsey 1988) are a natural starting point for NDRR valid inequalities. Specifically, if $\mathscr{A}'$ is a subset of arcs such that $\sum_{(i,j)\in\mathscr{A}'} q_{ij}^{km} > Q^{km}$ for some metric $m = 1, 2, \ldots, m^k$, then no feasible path for commodity k can contain all the arcs in $\mathscr{A}'$, and so the inequality $\sum_{(i,j)\in\mathscr{A}'} x_{ij}^{km} \leq |\mathscr{A}'| - 1$ is valid. However, this inequality is based solely on the arc weights and does not take into account a key requirement that governs the choice of commodity k's routing variables in the NDRR problem, namely, that the set of arcs on which commodity k flows must constitute an elementary path from $O(k)$ to $D(k)$. By exploiting this requirement, we can formulate a cut that is significantly stronger than the basic cover inequality. To illustrate this opportunity, consider the flow of commodity k on two arcs (i_1, j_1) and (i_2, j_2), and suppose the sum of the weights of these two arcs is less than the weight limit for every metric m. So, based solely on their individual weights, we cannot impose the requirement that at most

one of these arcs must be selected for commodity k. However, if we can determine that the two arcs cannot simultaneously belong to any feasible $O(k)$-to-$D(k)$ path, then the inequality $x^k_{i_1 j_1} + x^k_{i_2 j_2} \leq 1$ is valid. For this purpose, we note that any path that contains both arcs must either traverse arc (i_1, j_1) first before arc (i_2, j_2), or vice versa. For any metric m, the total weight of the shortest weight origin-to-destination path that contains arc (i_1, j_1) before arc (i_2, j_2) is $L1^{km} = L^{km}_{O(k)i_1} + q^{km}_{i_1 j_1} + L^{km}_{j_1 i_2} + q^{km}_{i_2 j_2} + L^{km}_{j_2 D(k)}$, whereas the smallest total weight if the order is reversed is $L2^{km} = L^{km}_{O(k)i_2} + q^{km}_{i_2 j_2} + L^{km}_{j_2 i_1} + q^{km}_{i_1 j_1} + L^{km}_{j_1 D(k)}$. The smaller of these two total weights is the weight of the shortest weight $O(k)$-to-$D(k)$ path that contains both arcs. Hence, if $Min\{L1^{km}, L2^{km}\} > Q^{km}$ for some metric m, we cannot route commodity k on any path that contains both arcs. In this situation, we say that $x^k_{i_1 j_1}$ and $x^k_{i_2 j_2}$ are *incompatible* flows, and the inequality $x^k_{i_1 j_1} + x^k_{i_2 j_2} \leq 1$ is valid. We can further strengthen this inequality by 'lifting' it, i.e., by adding some other flow variables to the left-hand side. For instance, if $\mathscr{A}' \in \mathscr{A}$ is a set of arcs such that the flow variables $x^k_{i'_1 j'_1}$ and $x^k_{i'_2 j'_2}$ for all $(i'_1, j'_1), (i'_2, j'_2) \in \mathscr{A}'$ are pair-wise incompatible with each other, then the inequality $\sum_{(i',j') \in \mathscr{A}'} x^k_{i'j'} \leq 1$ is valid.

We can also generalize this inequality. Let $r > 1$ be an integer, and arc set $\mathscr{A}' \subseteq \mathscr{A}$ with $|\mathscr{A}'| \geq r$. If no feasible solution to formulation [*NDRR*] permits commodity k to flow over more that $(r-1)$ arcs of $\mathscr{A}'$, then we say that $\mathscr{A}'$ is r-arc incompatible. Given an r-arc incompatible set $\mathscr{A}'$, let $\mathscr{A}'' \subseteq \mathscr{A} \setminus \mathscr{A}'$. If every arc $(i, j) \in \mathscr{A}''$ is incompatible for commodity k with every arc in $\mathscr{A}' \cup \mathscr{A}''$, then the inequality $\sum_{(i,j) \in \mathscr{A}'} x^k_{ij} + (r-1) \sum_{(i,j) \in \mathscr{A}''} x^k_{ij} \leq r-1$ is valid. This discussion illustrates how we can jointly exploit the weight constraints and the origin-to-destination routing requirement to develop tight valid inequalities for the NDRR problem.

Contingent Routing Inequalities This second class of inequalities further extends this principle of combining the weight constraints and the origin-to-destination routing requirement. A version of these inequalities, call *Lifted Turn* constraints, expresses the requirement that if a commodity flows on any arc in one subset, it must also flow on an arc of another subset. Consider a node v, where v is neither the origin nor the destination of commodity k. Let $Out(v) \subseteq \mathscr{N}_v^+$ be a subset of the outgoing arcs at node v. If $In(v) \subseteq \mathscr{N}_v^-$ denotes the maximal subset of incoming arcs (i, v) into node v such that every arc in $In(v)$ is pair-wise incompatible with every arc in $Out(v)$, then the inequality $\sum_{(i,v) \in In(v)} x^k_{iv} \leq \sum_{(v,j) \in \mathscr{N}_v^+ \setminus Out(v)} x^k_{vj}$ is valid. Note that the pair-wise incompatibility between arcs in $In(v)$ and arcs in $Out(v)$ arises because, for some m, $L1^{km} = L^{km}_{O(k)i} + q^{km}_{iv} + q^{km}_{vj} + L^{km}_{jD(k)} > Q^{km}$, for all $(i, v) \in In(v)$ and $(v, j) \in Out(v)$. Given the arc set $Out(v)$, we can identify $In(v)$ in the following way. Let $L^{km}_{min}(Out(v)) = \min_{(v,j) \in Out(v)}(q^{km}_{vj} + L^{km}_{jD(k)})$ and $L^{km}_{max}(\mathscr{N}_v^+ \setminus Out(v)) = \max_{(v,j) \in \mathscr{N}_v^+ \setminus Out(v)}(q^{km}_{vj} + L^{km}_{jD(k)})$. An arc (i, v) belongs to $In(v)$ if and only if $L^{km}_{O(k)i} + q^{km}_{iv} + L^{km}_{max}(\mathscr{N}_v^+ \setminus Out(v)) \leq Q^{km}$ and $L^{km}_{O(k)i} + q^{km}_{iv} + L^{km}_{min}(Out(v)) > Q^{km}$. BLM discuss an alternate way of identifying

the sets $In(v)$ and $Out(v)$, and show that the Lifted Turn inequality is facet defining under mild conditions. They also discuss a generalization of this inequality obtained by considering arcs that are incident to and from a subgraph spanning multiple nodes instead of the single node v.

Multicommodity Design Inequalities Both the previous two classes of inequalities restrict the flows of a single commodity and do not involve the design variables. The Multicommodity Design (*MCD*) inequalities impose variable upper bounds (that depend on the design variables) on the sum of flows of multiple commodities on various arcs. This very general class of inequalities is particularly effective in eliminating fractional LP solutions to [*NDRR*] because it relates the design and flow variables across multiple commodities and arcs. We first define two underlying commodity-specific relationships, called OR and IF relationships, that stem respectively from the Route Composition and Contingent Routing inequalities (both of which focus on a single commodity):

- an $OR(k, \mathscr{A}', \lambda)$ relationship specifies that no feasible path for commodity k can use more than λ arcs from a set $\mathscr{A}'$; and,
- an $IF(k, \mathscr{A}', \mathscr{A}'')$ relationship specifies that if commodity k flows on an arc of $\mathscr{A}' \subset \mathscr{A}$, then it must also flow on an arc of $\mathscr{A}'' \subseteq \mathscr{A} \setminus \mathscr{A}'$.

These two types of inequalities permit us to develop the following broad class of MCD inequalities. Let $\Omega = R_1, R_2, \ldots, R_Q$ denote Q relationships such that relationship q is either an $OR(k, \mathscr{A}', \lambda)$ relationship or an $IF(k, \mathscr{A}', \mathscr{A}'')$ relationship. Let I_{OR} and I_{IF} denote the subsets of indices q corresponding to the OR and IF relationships in the set Ω, and suppose δ_{ij} is an even number of relationships in Ω that involve arc $(i, j) \in \mathscr{A}$. Adding the inequalities in Ω to the forcing constraints $x_{ij}^{k_q} \leq y_{ij}$, and rounding down the resulting right hand side gives the following inequality for model [*NDRR*]:

$$\sum_{q \in I_{OR}} \sum_{(i,j) \in \mathscr{A}'_q} x_{ij}^{k_q} + \sum_{q \in I_{IF}} \sum_{(i,j) \in \mathscr{A}'_q} x_{ij}^{k_q} \leq \sum_{(i,j) \in \mathscr{A}} \frac{\delta_{ij} y_{ij}}{2} + \lfloor \sum_{q \in I_{OR}} \frac{\lambda_q}{2} \rfloor. \tag{8.8}$$

This MCD inequality tightens the [*NDRR*] if $\sum_{q \in I_{OR}} \lambda_q$ is odd. When the subset I_{IF} is empty, the MCD inequality has only OR relationships on the left hand side, and is facet defining under relatively mild conditions.

Composite Algorithm We can solve the NDRR problem by developing a tailored approach that uses the inequalities discussed above in a cutting plane approach, blended with an optimization-based heuristic. Since the number of inequalities in each of the classes is exponential in the input size, BLM use heuristics to identify inequalities violated by the LP solution. After solving the LP relaxation of this strong model, they use a LP-based heuristic to identify a feasible solution that fixes and releases variable values, an approach that is fast and effective for generating near-optimal solutions for large-scale instances. This method yields solutions that are within 1% of optimality, significantly outperforming (both in terms of solution

time and solution quality at termination) a standard branch-and-bound procedure (with built-in general cutting planes) that attempts to solve the base NDRR model without model strengthening and problem reduction.

3.3 Extension to Capacitated Network Design with Routing Restrictions

The approach discussed thus far can be extended to the Capacitated Network Design problem with Routing Restrictions (CNDRR) that not only imposes the routing constraints with non-bifurcated flow but also incorporates the following arc capacity constraints. If d^k denotes the demand for commodity k and u_{ij} is the capacity of arc (i, j), then we simply add the constraints $\sum_{k \in \mathcal{K}} d^k x_{ij}^k \leq u_{ij} y_{ij}$ for all arcs (i, j), to the [*NDRR*] formulation to model the CNDRR problem. Observe that, in this constraint although it suffices to use just the capacity u_{ij} as the right-hand side value, multiplying this value with the design variable y_{ij} strengthens the formulation. The CNDRR problem is more difficult because its LP relaxation can have fractional flows even when all the design variables are integer-valued and all the fractional flow paths for each commodity satisfy its routing requirements.

Researchers have developed various families of valid inequalities for the Capacitated Network Design (CND) problem (without routing restrictions) or its variant, the network loading problem (with discrete and modular capacities), to tighten the model. Such inequalities include the cutset, flow-cutset, partition, residual capacity, and c-strong inequalities. These inequalities remain valid even for the CNDRR problem, and so we can add them to the CNDRR formulation to strengthen it. Conversely, we can add the weight constraints and our NDRR valid inequalities to the formulation of the capacitated network design problem without routing constraints.

Interestingly, our commodity routing constraint and the arc capacity constraint are analogous but 'orthogonal' in the following sense. The routing constraint imposes an upper limit on the total weight of all the arcs on which a given commodity flows, whereas the arc capacity constraint limits the total flow of all the commodities that use a given arc. Conceptually, we can think of the routing constraint as a 'longitudinal' requirement along a commodity's path, whereas the capacity constraint is a 'lateral' requirement across commodities for an arc. We can exploit this complementary nature of the two requirements to tighten the valid inequalities for the routing requirements (or to eliminate variables) based on the capacity constraints and vice versa. We provide below some examples of such 'integration' of the two requirements to strengthen the CNDRR model.

- Suppose the demand d^k for a commodity k exceeds the capacity of an arc (i, j). In this case, not only can we omit the variable x_{ij}^k (since commodity k cannot flow on arc (i, j)), but also delete this arc from the network when computing the shortest weight paths needed to eliminate or fix flows of commodity k (see NDRR problem reduction methods in Sect. 3.1). Moreover, eliminating the arc flow variable x_{ij}^k can tighten both the Route Composition and Contingent Routing inequalities discussed in Sect. 3.2. For instance, for the Lifted Turn inequalities, although arc (i, j) may be compatible (from the perspective of the routing constraints) with one or more arcs in the set $In(v)$ incident to node $v = i$, we can omit this variable from the right-hand side of the inequality, thereby strengthening it. Conversely, if during problem reduction based on routing constraints, we discover that commodity k cannot flow an arc (i, j) (since the length of the shortest weight path through this arc exceeds a weight limit), then omitting this arc flow from the arc capacity constraint can help tighten any related CND valid inequalities.
- Consider two commodities $k1$ and $k2$ that can individually flow on an arc (i', j'), but cannot simultaneously on this arc due to the arc's capacity constraint, i.e., because $d^{k1} + d^{k2} > u_{i'j'}$. Further, suppose there is an arc (i_1, j_1) (and arc (i_2, j_2)) such that, if $k1$ (respectively, $k2$) flows on this arc it must necessarily flow on arc (i', j') to meet the weight limits, i.e., the length of the shortest weight path that includes arc (i_1, j_1) (respectively, (i_2, j_2)) but excludes arc (i', j') exceeds the weight limit for one or more metrics for commodity k_1 (respectively, k_2). In this case, either commodity $k1$ can flow on arc (i_1, j_1) or $k2$ can flow on (i_2, j_2), but not both, implying that the inequality $x_{i_1,j_1}^{k1} + x_{i_2,j_2}^{k2} \leq 1$ is valid. We can extend this inequality to subsets of three or more commodities. There are other such opportunities to develop 'integrated' inequalities that are based on jointly considering the arc capacity and routing requirements.
- We obtained the Multicommodity Design inequalities by aggregating judiciously chosen Route Composition and Contingent Routing inequalities, and applying rounding. For the CNDRR problem, we now have additional inequalities based on arc capacity constraints that we can consider for aggregation. For instance, based on the demand for different commodities and the capacity of an arc (i, j), we can impose cover inequalities of the form $\sum_{k \in \mathscr{K}'} x_{ij}^k \leq \lambda$, for an appropriate subset of commodities $\mathscr{K}'$. We can now consider combinations of these inequalities with those obtained using the routing requirements to develop an even richer set of Multicommodity Design inequalities.

In summary, for the CNDRR problem, we can strengthen the basic model by directly adding both our NDRR valid inequalities and cuts developed for capacitated network design problems. However, there are many opportunities to further reduce problem size and develop integrated inequalities based on the joint consideration of routing and capacity constraints.

4 NDRR Special Cases: Constrained Shortest Paths and Hop-Constrained Problems

Unlike the general NDRR problem, two special cases—the Constrained Shortest Path (CSP) and Hop-constrained Tree problems—have been well-studied in the literature. This section briefly reviews salient results and methods for these two special cases since the proposed modeling and solution strategies for these problems may prove useful for solving the broader NDRR problem. For instance, the CSP problem arises as a subproblem when solving the NDRR problem using column generation. The discussion also serves to illustrate approaches to develop and analyze approximation algorithms for the special cases, possibly pointing to principles that may extend to the general NDRR problem (for which no such analysis currently exists). Finally, for certain special cases (e.g., some hop-constrained problems with low hop limits), researchers have fully characterized the convex hull of feasible solutions. These results together with a hop-constrained problem formulation based on layered networks may provide the foundation to develop tighter (extended) NDRR problem formulations. Section 6 on Bibliographical Notes outlines the literature related to the topics discussed in this section and the next.

4.1 Constrained Shortest Path (CSP) Problem

The CSP problem is a single-commodity version of the NDRR problem that requires identifying the least expensive path from a given origin node s to a destination t whose total weight for each metric m does not exceed the corresponding weight limit. The literature sometimes refers to the problem containing only one routing constraint (one metric) as the *Budget-constrained Shortest Path* (BCSP) problem. To distinguish this problem from the more general version, we refer to the problem with a single commodity but multiple metrics and constraints as the *Weight-constrained Shortest Path* or *WCSP problem*. For these special cases, we can simplify the NDRR problem formulation as follows. Since there is only one commodity, we can omit the commodity index on the flow variables and weight limits. For the BCSP problem, since there is only one metric, we also omit the index m. Moreover, with positive costs, $y_{ij} = 1$ if and only if $x_{ij} = 1$. So, we can omit the design variables y_{ij} and forcing constraints (8.3), and use $(f_{ij} + c_{ij})$ as the flow cost of each arc (i, j) in the objective function. As noted previously, the CSP problem is NP-hard. We next discuss some theoretical results on approximation algorithms for the CSP problem, and later outline two interesting solution approaches that are effective in practice.

4.1.1 Approximation Schemes for the CSP Problem

For NP-hard problems such as the CSP problem, there are approximation (heuristic) algorithms that have provable bounds on solution quality. To facilitate the analysis of their performance, these algorithms are often simple and run in polynomial time. (For more complicated schemes such as neighborhood search, it is often not possible to characterize worst-case performance or even computational complexity.) Given any input or problem instance, an approximation scheme generates a feasible solution whose value is guaranteed (a priori) to be within a pre-specified (worst-case) factor of the optimal solution value. For minimization problems, this guarantee is expressed in terms of the maximum possible ratio of the cost of the approximate solution to the optimal value. Common techniques for obtaining these bounds include methods based on LP relaxation, Lagrangian relaxation, iterative rounding, randomized rounding, primal-dual methods, greedy heuristics, and scaling and rounding. The approach used depends on the problem's underlying structure and solution characteristics. For some problems, researchers have been able to develop desirable bounds that are either a constant factor (e.g., for network design special cases such as the Steiner tree, Traveling Salesman, and Facility Location problems) or depend on the problem dimensions. For instance, Balakrishnan et al. (1996) propose a efficient overlay heuristic for the uncapacitated network design problem and showed that this method yields a solution that is guaranteed to be within a factor of $|\mathscr{K}|$ of the optimal value (this is the first known bound for this problem). In other situations, the running time depends on the desired (maximum) approximation error $\epsilon > 0$. A fully polynomial-time approximation scheme, abbreviated as *FPTAS*, for a minimization problem generates a solution that is guaranteed to be within a factor of $(1 + \epsilon)$ of the optimal solution in running time that is polynomial in $1/\epsilon$ and the size of the input.

Since the specialized approximation algorithms rely on a problem's underlying structure to characterize worst-case performance, even seemingly minor changes to the problem affect the methods' applicability, analysis, and bounds. For instance, for the Knapsack problem, a slight variation of the greedy algorithm that selects items in decreasing order of value-to-weight is easy to analyze; it produces a solution value that is within a factor of $\frac{1}{2}$ of the optimal value. Although the BCSP problem has a knapsack-type constraint, the previous greedy approach is not applicable since the chosen arcs must also form an origin-to-destination path. The predominant method used to develop approximation schemes for the CSP problem is scaling-and-rounding. This approach entails reducing the weights or costs, by scaling and rounding, to low enough values so that the scaled problem can be solved efficiently. Although the scaled problem only yields an approximate solution to the original problem, the method has better time complexity than exact algorithms. The larger the scaling factor, the quicker the method runs but the solutions may be further from optimality. By judiciously selecting the scaling method and using other algorithmic steps (e.g., to determine tight bounds), the algorithm can yield an ϵ-optimal solution in polynomial time.

We next outline a FPTAS for the BCSP problem defined over acyclic graphs to illustrate these ideas. The method is based on a dynamic programming algorithm to solve the BCSP problem. When applied to the original problem (without any cost or weight scaling), this algorithm finds the optimal solution in $O(|\mathscr{A}|Z^*)$ time, where Z^* is the (unknown) optimal value of the problem. We can readily determine a priori upper bounds on Z^* (e.g., $Z^* \leq \sum_{(i,j)\in\mathscr{A}} c_{ij}$ or, better yet, the sum of the $(n-1)$ highest arc costs since no simple path can contain more than $(n-1)$ arcs). But since these bounds depend on the data (e.g., arc costs), the dynamic programming method, applied using the original parameters, is pseudo-polynomial. Now, suppose we can develop lower and upper bounds, LB and UB, on the optimal value such that $UB/LB \leq 2$. Then, for a specified approximation error ϵ, when we apply the dynamic program after scaling and rounding the arc cost coefficients to $d_{ij} = \lfloor c_{ij}/(LB\epsilon/(n-1))\rfloor$, the method runs in polynomial time ($O(|\mathscr{A}|n/\epsilon)$) and generates a solution whose approximation error is at most $\epsilon LB \leq \epsilon Z^*$, i.e., the solution is ϵ-optimal. To achieve the appropriate bounds needed for this approach, we start with $LB = 1$ and $UB =$ sum of the $(n-1)$ highest arc costs, and iteratively apply (in polynomial time) the scaling-and-rounding method to reduce the UB and raise the LB until $UB/LB \leq 2$. The method also extends to BCSP problems over general graphs. As this discussion illustrates, developing a FPTAS requires innovative approaches and insights about the problem structure and how to exploit its properties, with a focus on both characterizing the approximation error and reducing computational effort.

Unlike the BCSP problem, fewer approximation results are known for the more general WCSP problem with two or more weight constraints. Approximation algorithms are also available for a variant of the WCSP problem that allows bounded violation of the weight constraints. That is, in addition to approximating the objective function value to within a factor of $(1+\epsilon)$ (for minimization problems) these methods also permit relaxing (approximating) the weight constraints. When the weight limits can be exceeded by the same factor $(1+\epsilon)$, the approximation algorithm for the WCSP problem essentially seeks an appropriate solution(s) to a multiobjective shortest path problem having costs and routing metrics as different criteria.

4.1.2 CSP Solution Algorithms

We next discuss two interesting solution methods for the CSP problem (with non-negative arc costs) that exploit its special structure. These methods, although not polynomial, are effective in practice.

4.1.3 Handler and Zang's Algorithm

For the BCSP problem, Handler and Zang (1980), abbreviated as HZ, consider the Lagrangian relaxation obtained by dualizing the single weight constraint with multiplier u, and proposed a novel solution approach to solve the Lagrangian dual and close the optimality gap. For this scheme, the Lagrangian subproblem:

$$L(u) = \min \sum_{(i,j)\in\mathscr{A}} (c_{ij} + uq_{ij})x_{ij} - uQ, \qquad \text{subject to (8.2) and (8.5)},$$

is a shortest path problem using arc lengths $(c_{ij} + uq_{ij})$; this path's length, when reduced by uQ, represents the optimal value $L(u)$ of the Lagrangian subproblem. For any $u \geq 0$, $L(u)$ is a lower bound on the optimal value of the original problem. We can solve the Lagrangian dual problem, maximize $\{L(u){:}u \geq 0\}$, by iteratively adjusting u using, for instance, a general technique such as sub-gradient optimization or a more specialized approach. HZ propose a tailored method that exploits the BCSP problem's special structure (and the fact that we need to optimize just one dual multiplier) to solve the Lagrangian dual and reduce any remaining duality gap. The Lagrangian value $L(u)$ is a piecewise linear, concave function of u, with each segment of the piecewise function corresponding to the Lagrangian value for one origin-to-destination path. Starting with two paths (one which minimizes cost without the budget constraint and one which minimizes budget usage), the dual solution method iteratively refines an upper (piecewise linear) approximation for the $L(u)$ function by sequentially generating s-to-t paths and updating the multiplier u. The method monotonically increases the Lagrangian lower bound, denoted as LB. If at any iteration, the Lagrangian solution is feasible (i.e., satisfies the budget constraint), we can also update the upper bound UB if the cost of this new path is lower than the current best upper bound. When the method terminates, we may still have a duality gap, i.e., LB may be less than UB. The following approach closes this gap. For the final value of the dual multiplier u, instead of finding just the shortest path (as we do to solve the Lagrangian subproblem), suppose we sequentially identify the rth shortest path (using the Lagrangian costs), for increasing values of r. Let $L_r(u)$ be the (Lagrangian) cost of the rth shortest path. We update LB as $L_r(u)$, and can possibly update UB if the rth shortest path is feasible for the BCSP problem. We increment r and repeat the process until LB equals or is sufficiently close to UB. Since the network contains only a finite number of (elementary) origin-to-destination paths, this gap reduction procedure will terminate in a finite number of iterations.

Node Labeling Approach

Another approach to solve the BCSP problem is by generalizing Dijkstra's label-setting shortest path algorithm. The generalization entails associating multiple labels with each node i, one for each sub-path from node s to i that can potentially belong to the optimal solution. With one routing constraint, each label contains two elements, the cost and weight, associated with a path from s to i. We only maintain

labels for paths that are undominated, i.e., if p and p' are two different paths from s to i, and path p' has higher cost than path p, then this path is undominated only if it has strictly lower weight than path p. We can also omit some labels based on feasibility requirements: if the path corresponding to a label cannot be extended to reach node t within the weight limit (i.e., the path is not a sub-path of a feasible route for the commodity), then we can ignore this path. The method initializes the problem by assigning the label (0, 0) to node s, and then iteratively chooses the lowest cost label among all labels that have not been previously chosen. Since each node can have up to Q (the weight limit) labels (assuming nonnegative integer weights), the node labeling approach may not be effective when Q is large. Preprocessing techniques can significantly improve the empirical computational performance of the node labeling algorithm. These techniques consist of feasibility tests to identify and delete nodes and arcs that an optimal solution will not use. Using information from a Lagrangian relaxation of the weight constraint, e.g., to prune node labels, yields further improvements. In extensive computational tests, the node labeling algorithm with preprocessing is more effective than scaling techniques (Sect. 4.1.1). The node labeling approach can be extended to the WCSP problem with multiple weight constraints.

To conclude, in the context of the NDRR problem, the CSP problem is interesting and relevant because: (1) it captures the core NDRR feature of finding an origin-to-destination path for each commodity subject to routing constraints; (2) the CSP problem has received significant attention in the literature on approximation algorithms since the single commodity structure makes it more tractable; and (3) the CSP problem arises as a subproblem when we consider decomposition algorithms such as Lagrangian relaxation and column generation for solving the general NDRR problem (see Sect. 5). The CSP approximation algorithms and analysis may provide leads for analyzing the worst-case performance of approximation methods for the NDRR problem, although the presence of shared fixed costs, across commodities, in the NDRR problem may significantly complicate the analysis (possibly accounting for the lack of analogous results on network design problems, in general). The CSP algorithms, particularly the node labeling approach, can serve to solve subproblems quickly in NDRR decomposition approaches. We note that the problem reduction methods and two classes of valid inequalities—the Route Composition and Route Coordination inequalities—that BLM developed for the general NDRR problem also apply to the WCSP problem. With these inequalities, solving the strengthened WCSP model using state-of-the-art integer programming solvers may also be competitive. To our knowledge, this approach has not been tested.

4.2 Hop-Constrained Routing and Design Problems

We now consider the special version of the commodity routing requirement in which all arc weights are equal to one. If H^k denotes commodity k's weight limit in this constraint, the routing requirement states that each commodity k must use a path that

contains no more than H^k arcs (hops). Therefore, we refer to this restriction as a *hop constraint* and to the corresponding weight limit as the hop-limit. (More generally, if all arcs have the same weight, not necessarily one, we can scale the weights and scale and round down the weight limit to convert the constraint to a hop constraint.) We refer to this special case of the NDRR problem with only hop restrictions as the *Hop-constrained Network Design* (HCND) problem. Balakrishnan and Altinkemer (1992) were the first to study the HCND problem; they develop and test a solution procedure based on Lagrangian relaxation. The literature has largely focused on a restricted version that we call the Hop-constrained Tree (HCT) problem in which there is a single source or root node that needs to be connected to other specified nodes, called terminal nodes, via a *tree* network, and all routing costs are zero. The problem is typically defined over an undirected network, and assumes the same hop-limit H for all commodities. If all nodes of the network, except the root node, are terminal nodes, then the required configuration is a spanning tree, i.e., the problem is a Hop-constrained Minimum Spanning Tree (HCMST). Otherwise, the design is a tree that spans the root node and all terminal nodes, and optionally includes non-terminal, i.e., Steiner, nodes. We refer to this latter problem as the Hop-constrained Steiner Tree problem. This section discusses approximation schemes for the HCMST problem, a layered network representation of hop-constrained path and tree problems that yields extended (tighter) model formulations, and some polyhedral results for these problems.

4.2.1 Approximation Algorithms for the HCMST Problem

Approximation algorithms for constrained tree problems largely focus on the Diameter Constrained Minimum Spanning Tree (DCMST) problem which, as we discuss next, is related to the HCMST problem. Given a maximum permitted diameter D, the DCMST problem seeks a minimum cost spanning tree such that the number of edges (hops) between any two pairs of nodes is at most D. If the diameter is even, say, $D = 2H$, we can solve n HCMST problems, each with a different node as the root node and hop-limit equal to H, and pick the lowest cost solution among these n problems as the optimal configuration for the DCMST problem. If D is odd and equals $(2H + 1)$, then any feasible DCMST solution must have an edge (i, j) such that every other node is connected to either node i or node j via a path containing no more than H edges. Thus, if we merge (or contract) nodes i and j, the solution is a DCMST with even diameter $(D - 1) = 2H$ (with the merged node as its center). So, we can solve the original DCMST problem by solving $|\mathscr{A}|$ HCMST problems, each obtained by contracting one edge of the original network. Conversely, we can also transform a HCMST problem into an equivalent DCMST problem as follows. Given the root node s and hop-limit H of the HCMST problem, we augment the network by adding two strings of H nodes, incident from node s, connected by zero cost arcs. Then, solving a DCMST problem, with diameter limit $D = 2H$, over the augmented network yields the HCMST solution rooted at node

s and satisfying the hop-limit. So, given an approximation algorithm with known performance guarantee for the DCMST problem, we can obtain an approximate solution with the same guarantee for the HCMST problem. The DCMST problem is NP-Hard even with $D = 4$ and with edge costs that are all either one or two (Garey and Johnson 2002), motivating the exploration of approximation methods. For instance, for the Diameter Constrained Steiner Tree problem (a generalization of DCMST in which the solution is only required to span a subset of nodes called terminal nodes and can optionally span other nodes called Steiner nodes), an approximation algorithm combining greedy selection and exhaustive search has a worst-case ratio of $O(\log(|T|))$, where T is the set of terminal nodes.

Interestingly, the HCMST special case with $H = 2$, which we call the two-hop HCMST problem, is equivalent to the uncapacitated facility location (UFL) problem. Given an instance of the UFL problem (with a dummy source node which is connected to all facility nodes), we can construct an equivalent two-hop HCMST instance by adding zero cost arcs between the facility nodes. Conversely, we obtain a UFL instance (with the root node as the dummy source node) corresponding to any two-hop HCMST instance by defining both a facility and a customer for each original non-root node, and assigning the cost for each original arc (i, j) to the arc from plant i to customer j (this cost is zero when $i = j$). These transformations imply that the approximation results for the UFL problem, such as the constant worst-case bounds based on greedy and cost scaling methods, also apply to the two-hop HCMST problem with metric costs. Developing a constant bound algorithm for the general HCMST problem remains an open problem.

4.2.2 Polyhedral Results for Hop-Constrained Path Problems

In Sect. 3, we discussed some polyhedral results for the general NDRR problem. For NDRR special cases when the underlying flow problem is a hop-constrained path or tree problem, there are specialized valid inequalities and polyhedral results for the underlying flow problems (assuming that all origin-to-destination paths must be elementary).

For the Hop-constrained Shortest Path (HCSP) problem, it is possible to characterize the underlying polytope when the hop-limit H is small and fixed. Since the HCSP problem has only one commodity, we omit the commodity index k in the following discussions. For $H = 2$, the solution can only contain arcs of the type (s, i) or (i, t) for some node $i \in \mathcal{N}$. Therefore, we can set the flow $x_{ij} = 0$ on all other arcs (i, j) that are not incident at either node s or t. These equalities, together with the flow conservation constraints for nodes s and t, and nonnegativity requirements on the x_{ij} variables, completely describe the convex hull of feasible solutions to the HCSP problem with $H = 2$. For $H = 3$, the flow conservation constraints at all nodes, the nonnegativity requirements on the full set of x_{ij} variables, and the inequalities

$$x_{si} - \sum_{j \in \mathcal{N} \setminus \{s,t\}} x_{ij} \geq 0 \qquad \forall i \in \mathcal{N} \setminus \{s,t\} \tag{8.9}$$

together give a complete description of the HCSP polytope.

Constraint (8.9) is a special version of a broad class of inequalities called *jump* inequalities. The basic jump inequality has the following structure. Let $V_1, V_2, \ldots, V_{H+2}$ be pairwise node-disjoint sets that partition the node set $\mathcal{N}$, with $V_1 = \{s\}$ and $V_{H+2} = \{t\}$. Define jump $J = \cup_{1 \leq i \leq j-2} \{V_i, V_j\}$, where $\{V_i, V_j\}$ is the set of arcs (a, b) such that $a \in V_i$ and $b \in V_j$. If $J(s\text{-}t, H)$ denotes the set of all jumps, then the jump inequality is

$$\sum_{(a,b) \in J} y_{ab} \geq 1 \qquad \forall J \in J(s\text{-}t, H) \tag{8.10}$$

By definition of the jump J, if an s-to-t path does not use any of the arcs in J, then the path must have at least $(H + 1)$ arcs, and so is not a feasible path for the hop-constrained problem. Lifted versions of these jump inequalities can define facets for HCSP problems with higher hop-limits ($H > 3$).

4.2.3 Layered Networks and Extended Formulations for Hop-Constrained Problems

When the routing requirement is a hop constraint, defining the commodity flows over a *layered* (expanded) network provides a convenient and intuitive representation of the network design (or shortest path) problem and also yields tighter formulations. We first discuss the structure and properties of the layered network for a single commodity (e.g., for the HCSP problem), and then address extensions to problems with multiple commodities, including hop-constrained tree problems. An important by-product of these layered network representations is that, using projection techniques on the associated extended formulations, we can obtain strong valid inequalities in the original space of design variables.

Layered Network Representation for Hop-Constrained Paths

Suppose a commodity from s to t must be routed on a path containing no more than H arcs (in the following discussion, we omit this commodity's index). The original NDRR problem formulation [*NDRR*] defines (binary) commodity flow variables x_{ij} on the original network, and imposes the hop constraint as $\sum_{(i,j) \in \mathcal{A}} x_{ij} \leq H$. Instead, suppose we define the flow variables over the following expanded network containing $(H + 1)$ layers, indexed from $h = 1$ to $h = (H + 1)$. Layer 1 contains only the source node s, and layer $(H + 1)$ only the sink node t. Each intermediate layer, $h = 2, 3, \ldots, H$, contains a copy of every node $i \neq s$, labeled as node $< i, h >$. The source node has label $< s, 1 >$, and sink node in the last layer has label $< t, H + 1 >$. If the original graph contains an arc (i, j), in the layered graph we connect node $< i, h >$ to node $< j, h + 1 >$, except when h is H we only consider

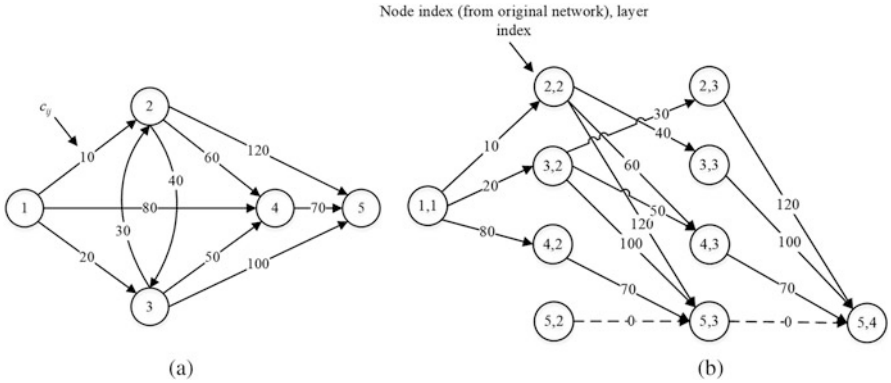

Fig. 8.2 Example of layered network for one commodity. (**a**) Original network. (**b**) Layered network with $H = 3$

$j = t$. We also add 'dummy' arcs, with zero cost, from $< t, h >$ to $< t, h + 1 >$ for $h = 2, 3, \ldots, H$. Since the hop limit H must be less than the number of nodes n (assuming that we only permit elementary paths, which holds if all costs are non-negative), the size (number of nodes and arcs) of the layered network is no more than n times the size of the original network. Figure 8.2 illustrates this construction. Figure 8.2a shows the original network with source node $s = 1$ and sink node $t = 5$. Figure 8.2b shows the layered network with hop-limit $H = 3$.

From this construction, we can readily see that any path from $< s, 1 >$ to $< t, H + 1 >$ in the layered network satisfies the hop constraint, and conversely every feasible path in the original network has a corresponding path from $< s, 1 >$ to $< t, H+1 >$ in the layered network. Therefore, if we define the routing variables as arc flows over the layered network (instead of the original network), then we do not need to explicitly impose the hop constraints on the routing variables. Specifically, instead of using the flow variables x_{ij} defined over the original graph, we now define disaggregated *hop-indexed* flow variables x_{ij}^h, for $h = 1, 2, \ldots, H$. The variable x_{ij}^h takes the value one if arc (i, j) is the h^{th} hop on the commodity's route from origin s to destination t, and is zero otherwise. Equivalently, x_{ij}^h is the flow from node $< i, h >$ to node $< j, h + 1 >$ in the layered graph. The flow conservation constraints are:

$$\sum_{j \in \mathcal{N}_i^+} x_{ij}^h - \sum_{j \in \mathcal{N}_i^-} x_{ji}^{h+1} = \begin{cases} 1 \text{ if } i = s, \\ -1 \text{ if } i = t, \\ 0 \text{ otherwise.} \end{cases} \quad \forall\, h = 1, 2, \ldots, H-1. \tag{8.11}$$

For the HCSP problem, since any elementary s-to-t path cannot contain an arc (i, j) on more than one of the H hops, the 'routing' cost associated with each disaggregated flow variable x_{ij}^h is the same as the routing cost c_{ij} on the original arc (i, j). So, the HCSP problem's layer-indexed formulation, denoted as [*L-HCSP*],

minimizes $\sum_{(i,j)\in\mathscr{A}} \sum_{h=1}^{H} c_{ij} x_{ij}^h$ subject to the flow conservation constraints (8.11) and integrality requirements $x_{ij}^h = 0$ or 1 for all $(i, j) \in \mathscr{A}, h = 1, 2, \ldots, H$. This model is the same as the formulation for the (unconstrained) shortest path problem over the layered network. Indeed, we can relate the HCSP problem's interpretation as the shortest path in the layered network to the well-known dynamic programming recursion: $d(j, h) = \min\{d(j, h-1), \min_{i:(i,j)\in\mathscr{A}}\{d(i, h-1)+c_{ij}\}\}$, where $d(j, h)$ denotes the shortest distance from node s to node j using h or fewer hops, for solving the hop-constrained shortest path problem (e.g., Lawler 1976). These observations imply that formulation [*L-HCSP*] is exact, i.e., it completely describes the HCSP polytope. In contrast, the LP relaxation of the original NDRR formulation, specialized to the HCSP problem, can have fractional solutions with non-zero integrality gap. Thus, although the layered formulation contains H (which is $O(n)$) times as many variables as the original formulation [*NDRR*] applied to the HCSP problem, the use of hop-indexed flow variables serves to strengthen the LP relaxation and close the integrality gap.

Layered Network and Extended Formulation for Hop-Constrained Trees
As noted earlier, the HCT problem requires designing a minimum cost tree that connects a root node s to a specified set T of terminal nodes, with a hop-limit of H on each root-to-terminal path. When T includes all nodes except the root node, the design is a spanning tree; otherwise, it is a Steiner tree. We can view this problem as a tree-constrained multicommodity HCND problem containing one commodity k for each terminal node that originates at the root node s and has node $k \in T$ as its destination. Applying the previous hop-indexed disaggregation to each commodity's flow variables, we obtain a formulation that is stronger than the [*NDRR*] formulation with added tree constraints.

4.2.4 Extended Formulations for General NDRR Problems

The models based on layered networks and the hop-indexed variables discussed in the previous section add to the rich history of using disaggregate variables to improve the effectiveness of formulations. Another related development is the use of extended formulations, obtained by adding new variables, for various combinatorial optimization problems. Research on this topic of extended formulations started mainly as a theoretical tool, but in recent years researchers have also examined their value for strengthening the LP relaxation when solving difficult integer programs (see Wolsey 2011). We can interpret the formulations based on the layered network representation of hop-constrained problems as extended formulations for the underlying problem.

The layered network concepts and modeling enhancements can also extend to more general NDRR problems, e.g., with multiple sources and destinations, without explicit tree configuration requirements, and with general (and multiple) routing constraints (vs. hop limits). For instance, we can represent problems with a general weight constraint (with arbitrary positive integer coefficients) in the following way.

Assume, for notational simplicity, that the integer arc weights and weight limits are the same for all commodities. For a routing constraint with a limit of Q for commodity k, we have $(Q + 1)$ instead of $(H + 1)$ layers in the layered network representation. Arcs between layers are not always between adjacent layers as is the case for the hop-constrained version. Instead, for each arc (i, j) with weight q_{ij} in the original network, the layered network contains an arc from node $< i, q >$ to node $< j, q + q_{ij} >$ for $1 \leq q \leq (Q - q_{ij} + 1)$. Corresponding to each such arc, we define a disaggregated flow variable x_{ij}^{kq}, and use flow conservation constraints analogous to (8.11) and the forcing constraints $\sum_{q=1}^{Q-q_{ij}+1} x_{ij}^{kq} \leq y_{ij}$. Further, if the solution is required to be a tree, then we can also disaggregate the design variables as long as the arc weights and weight limits are the same for all commodities. Note that the size of the layered network, and hence the number of variables in the disaggregate formulation, is pseudo-polynomial since it depends on the weight limit Q. When this limit is very large or if there are multiple routing constraints, the extended formulation will be too large to solve directly using general purpose integer programming solvers. To mitigate this difficulty, we can apply an iterative approach that starts with a small layered network, and increases its size until we obtain a feasible solution that is sufficiently close to optimal.

If the problem imposes more than one routing constraint for each commodity, we can develop a 'multi-dimensional' layered network that implicitly captures multiple routing requirements. For instance, suppose there are two routing constraints with weight limits Q^1 and Q^2. Then, the multi-dimensional layered network contains $Q^1 Q^2$ layers containing nodes of the form $< i, q^1, q^2 >$, with $1 \leq q^m \leq (Q^m + 1), m = 1, 2$, and arcs from this node to node $< j, q^1 + q_{ij}^1, q^2 + q_{ij}^2 >$ for every arc (i, j) of the original network. Using the flow variables (and design variables, for tree sub-networks) on this layered network, we can develop an extended formulation for the NDRR problem.

In summary, for network design problems with hop limits or general weight constraints, layered network representations permit incorporating these routing restrictions implicitly by defining appropriate disaggregated decision variables instead of explicitly adding routing restrictions in the basic NDRR problem. This approach yields tighter formulations, but at the expense of requiring many more decision variables. Note that these extended formulations also apply to the capacitated version of the problem, e.g., the CNDRR problem with hop limits as the routing restrictions. For this problem, in each arc capacity constraint, we replace the original arc flow variables with the sum of the corresponding layer-indexed variables. Moreover, we can also add cuts developed for capacitated network design (using the sum of disaggregate flow variables in place of original arc flow variables) to this model to further strengthen the model. Since the layered formulation is tighter than the previous CNDRR model, it can improve the performance of the decomposition schemes discussed next.

5 Decomposition Strategies for the NDRR Problem

Section 3 discussed BLM's cutting plane approach to solve the general NDRR problem, and Sect. 4 outlined an alternate approach, using extended formulations, to strengthen the problem's LP relaxation and accelerate integer programming solvers. This section outlines other possible solution strategies based on decomposition methods, including Lagrangian relaxation, column generation, and Benders decomposition.

5.1 Lagrangian Relaxation

Lagrangian relaxation techniques are attractive when the problem contains embedded special structures that can be solved more easily. The NDRR problem has two such structures, corresponding respectively to uncapacitated network design and constrained shortest paths. If we dualize the routing constraints (8.4), the resulting subproblem is an uncapacitated fixed charge network design problem. Although this problem is NP-hard, Balakrishnan et al. (1989) describe a dual ascent procedure that is very effective in solving problems of reasonable size. When used as the procedure to solve Lagrangian subproblems, it may be possible to further accelerate the procedure by warm-starting it using dual values from the previous iteration. However, identifying a solution to the original NDRR problem that satisfies all the routing constraints may be difficult (recall from Sect. 2 that, even if we are given the network design, finding feasible solutions can be NP-hard when there are multiple routing constraints).

An alternative Lagrangian relaxation scheme consists of dualizing the forcing constraints (8.3), resulting in $|\mathscr{K}|$ CSP subproblems, one for each commodity k. CSP solution methods such as the node labeling algorithm (Sect. 4.1.2) are quite effective and quick. Potentially, when repeatedly solving CSP problems for a commodity, each of which differ only in the arc cost coefficients (which depend on the Lagrangian multiplier values), we can modify these methods to improve their performance (e.g., previous feasible solutions yield upper bounds for the current problem). Of course, since the computational effort for these pseudo-polynomial algorithms depends on the weight limits, solving each subproblem can be time consuming when these limits are large. Also, with a large number of arcs and commodities, the number of forcing constraints, and hence Lagrangian dual variables, is very large; so, the convergence of the Lagrangian dual problem may be slow.

We note that neither subproblem in the above two Lagrangian relaxation schemes satisfies the integrality property (i.e., the optimal solution to the LP relaxation of the subproblem may have fractional values). So, the best Lagrangian lower bound can exceed the LP lower bound, obtained by solving the LP relaxation of the original NDRR problem formulation (8.1)–(8.5). So, for problem instances where using

either of the above subproblem solution procedures is practical, using Lagrangian relaxation at intermediate nodes of a branch-and-bound procedure can outperform standard LP-based branch-and-bound algorithms.

For the CNDRR problem, we have additional choices for the Lagrangian scheme. For instance, dualizing the arc capacity constraints results in NDRR subproblems, whereas relaxing the routing restrictions yields capacitated network design subproblems. To further simplify the subproblems, we can dualize additional constraints such as the forcing constraints or the flow conservation equations to obtain CSP or knapsack subproblems. In general, capacitated design problems tend to be more difficult to solve using this technique unless the model is further strengthened with valid inequalities (such as design inequalities based on arc capacity constraints).

5.2 Column Generation (Dantzig-Wolfe Decomposition)

As noted in Sect. 2, instead of formulating the NDRR problem using arc flow variables, we can also consider a path selection formulation that uses path flow variables corresponding to feasible origin-to-destination paths (satisfying the routing restrictions) for each commodity. We can view this reformulation (from the arc to path representation) as a change of variables, just as the layered network representation replaces the original arc flow variables with layer-indexed arc flow variables. Since the path flow model only considers origin-to-destination paths that meet the routing constraints, it has a stronger LP relaxation. However, since the number of such paths is exponential in the network size, we cannot explicitly solve the full model (except when the number of candidate paths is limited due to highly restrictive weight or hop limits). Instead, we can apply a column generation technique that iteratively generates promising paths based on the LP dual solution, embedded in a branch-and-price procedure to close the integrality gap. Applying column generation to the general NDRR problem requires iteratively solving a CSP subproblem for each commodity k in order to find a feasible $O(k)$-to-$D(k)$ path with negative reduced cost (not surprisingly, these pricing problems are the same as the Lagrangian subproblems that we need to solve when we dualize the forcing constraints of the arc flow model). The restricted master problem (RMP) is a linear program that chooses the path to be used for each commodity k from among the paths generated so far. The optimal dual prices of the RMP determine the arc costs in the CSP subproblems. Branching is needed to reduce the optimality gap between the LP value and the current best upper bound.

For the NDRR path flow model, the column generation procedure can require excessive number of iterations to converge because of the huge number of candidate paths for each commodity. One cause of slow convergence is that, during the initial iterations, the RMP (with only few columns) may not approximate the full problem very well and so its solution may not be close to optimality for the full problem. Consequently, the dual prices at these iterations may not adequately approximate the optimal dual prices of the full problem, causing the procedure to generate new

columns that are not very useful. These columns can cause the RMP solutions to vary widely from iteration to iteration. Stabilization techniques can improve the convergence of column generation procedure by mitigating these difficulties. One successful approach has been to use a pre-chosen stabilization point, and constrain the RMP solutions to remain "close" to the stabilization point through the use of penalty functions.

Another technique is to improve the LP bounds of column generation model by adding cuts to the master problem. This technique, known as branch-and-cut-and-price, can be useful but care is necessary since the added cuts introduce new dual variables that change the structure of the pricing operation. If the pricing subproblem becomes difficult or intractable, then the column generation approach is impractical. However, for specific types of added cuts known as robust cuts, the basic structure of the pricing operation is unchanged. For example, in the arc-path approach to column generation, cuts using only design variables are robust.

The column generation approach also applies to the CNDRR problem, except that the RMP now also includes the arc capacity constraints (expressed in terms of the path flow variables). Again, it is important to strengthen the master problem by adding valid inequalities, preferably using robust cuts so as not to complicate the subproblems. Column generation has proven to be among the most successful methods for related capacitated problems such as vehicle routing and Capacitated Minimum Spanning Tree (CMST) problems. The CMST problem requires find the minimum cost spanning tree such that the total demand in each subtree of a designated root node does not exceed the capacity C of the arcs incident from the root node. For this problem, instead of paths, the columns represent feasible subtrees (that satisfy the capacity constraint) with degree one at the root node. Unfortunately, the pricing subproblem is strongly NP-hard and not practical computationally, necessitating some improvements to the branch-and-price approach such as modifying the column space. Specifically, instead of subtrees, using a set of arborescence-like structures permits a pseudo-polynomial algorithm (with respect to C) for the pricing subproblem. We can build upon these methods to solve CNDRR variants such as CMST with hop or more general routing restrictions. For instance, we can use a layer-indexed model for the pricing subproblem to capture both the hop limits and capacity limit for arcs incident to the root node. Such extensions provide fertile ground for further work.

5.3 Benders Decomposition

Benders decomposition entails fixing a subset of variables in a master problem, and solving LP subproblems whose dual values induce so-called Benders cuts in the master problem. Modern implementations of this approach, known as Benders branch-and-cut, embed the procedure into a branch-and-bound framework, add a Benders cut at each node in the search tree, and solve the master problem only once.

For the NDRR problem, if we fix the design variables at values $y_{ij} = \overline{y}_{ij}$ in the master problem, the resulting subproblems are CSP problems for each commodity k over the candidate network formed by $\overline{\mathscr{A}} = \{a \in \mathscr{A} | \overline{y}_{ij} = 1\}$. Unfortunately, the resulting subproblems are integer programs, and so Benders decomposition is not directly applicable. However, for the special case when all the routing constraints (8.4) are hop-limits, by using the disaggregated (hop-indexed) flow variables, the Benders subproblems are linear programs since they are simply (unconstrained) shortest path problems over the layered network for each commodity (but only including arcs chosen by the master problem). The Benders subproblems can, however, be infeasible because, in the current design chosen by the Benders master problem, the destination $D(k)$ may not be reachable from the origin $O(k)$ within the desired hop-limit (i.e., $O(k)$ and $D(k)$ are not connected in the layered network). In this case, we must generate and add a Benders feasibility cut to the master problem. In general, it is difficult to generate effective Benders feasibility cuts; this topic is currently an active area of research. Another possible strategy is to skip adding an feasibility cut and continue the branching process. This approach trades off the additional effort required in the search tree with the benefit of not generating feasibility cuts.

As we noted in Sect. 4.2.4, for NDRR problems with one routing constraint, having non-unitary coefficients, we can use the pseudo-polynomial layered network to model the disaggregate flow variables. In this case too, the subproblem of selecting a feasible path for a given set of design variables is a linear program, permitting the application of Benders decomposition.

Finally, the Benders approach also extends to CNDRR problems, except that feasibility of subproblems now depends on both whether the design contains at least one feasible origin-to-destination path (satisfying routing restrictions) for every commodity, and also meets arc capacity constraints (across all commodities). We can potentially improve Benders performance by tightening the LP relaxation of the design problem formulation in the master problem by adding cuts that only involve the design variables, such as those obtained using projection techniques of the capacitated network design problem, or reformulating the problem using disaggregated variables. Note that adding cuts to the Benders master problem is analogous to adding robust cuts for column generation. Master problem cuts (robust cuts) do not complicate the solution of the Benders subproblems (column generation pricing operation). Adding more general cuts can complicate the efficient solution of the subproblems just as non-robust cuts complicate the column generation pricing operation.

6 Bibliographical Notes

Valid Inequalities and Cutting Plane Methods for the General NDRR Problem Although network design problems have been studied extensively (e.g., Magnanti and Wong 1984; Balakrishnan et al. 1997; Crainic 2000), little research has been

done on the NDRR problem. Barnhart and Schneur (1996), Armacost et al. (2002), and recently, Yildiz and Savelsbergh (2019) study optimal design for express delivery using ground and air transportation in order to determine multimodal time-sensitive origin-destination routes. Other related network design problems that also have a flavor of network design and/or flow routing with restrictions include reliable path routing, reliable network design, and survivable network design (e.g., Balakrishnan et al. 2009). Balakrishnan et al. (2017) (BLM) is the first paper to address the general NDRR problem, and develop a tailored cutting plane-based approach for this problem. The discussion in Sect. 3 is largely based on this paper. BLM provides a more general framework and treatment of the three classes of valid inequalities discussed in Sect. 3. Moreover, they describe some sophisticated lifting procedures, and prove that some versions of these inequalities are facets of the NDRR polyhedron. They report extensive computational results from applying the cutting plane procedure (using heuristic separation procedures to iteratively find violated inequalities) at the root node of a branch-and-bound algorithm, combined with an optimization-based heuristic method, for a variety of NDRR test problems containing up to 80 nodes, 320 arcs, and 240 commodities.

Extension to Capacitated NDRR Problems Magnanti et al. (1993, 1995) were among the first to study the capacitated network design problem where multiple modular facilities can be installed on the network arcs. They developed the cutset, residual capacity and 3-partition inequalities, and studied their theoretical and computational effectiveness. Bienstock and Gunluk (1996) developed several facet-defining inequalities that extend the cutset and the 3-partition inequalities. Atamturk and Rajan (2002) study single-arc set relaxations of the problem and show that the separation problem of the residual capacity inequalities (for the splittable case) can be solved in linear time while the separation problem for the c-strong inequalities (developed by Brockmuller et al. (2004) for the unsplittable case) is NP-hard. They extend the c-strong inequalities and conduct computational experiments to test the effectiveness of these inequalities. Benhamiche et al. (2016) study the polyhedral structure of a model where a commodity flow cannot split even across two different facilities on the same arc. Gendron et al. (1999) provide a survey of multicommodity capacitated network design models. They also summarize the theoretical strengths and present a computational comparison of several different relaxations of an arc-based formulation of the problem.

Approximation Schemes for the Budget-Constrained Shortest Path (BCSP) Problem Vazirani (2013) and Williamson and Shmoys (2011) provide comprehensive discussions of approximation schemes for various problem settings. For the BCSP problem over acyclic graphs, Warburton (1987) was the first to develop a FPTAS. The complexity of his approximation scheme is $O(n^3\epsilon^{-1}\log(n)\lceil\log(UB)\rceil)$, where ϵ is the performance guarantee (i.e., the heuristic solution value is within a factor of $(1+\epsilon)$ of the optimal solution value) and UB is an upper bound on the optimal solution value. Hassin (1992) employs the principles underlying this method, but uses a constant bound on the ratio of UB to LB to develop an approximation algorithm with complexity $O(|\mathscr{A}|n^2\epsilon^{-1}\log(n\epsilon^{-1}))$. Section 4.1.1

summarizes this method. Lorenz and Raz (2001) further improve the approximation scheme, achieving a n-fold reduction in time complexity, by simplifying the method for obtaining upper and lower bounds, and permitting a larger approximation error in the first stage. This method runs in $O(|\mathscr{A}|n(\log\log n + 1/\epsilon))$. Ergun et al. (2002) also improve Hassin's algorithm but by making the scaling factor adaptive, starting with a large scaling factor. As the difference between current upper and lower bounds decreases, the method reduces the scaling factor. This strategy improves solution quality without adversely affecting the running time, resulting in a FPTAS with time complexity of $O(|\mathscr{A}|n\epsilon^{-1})$.

CSP Solution Algorithms One possible drawback to the Handler-Zang (*HZ*) approach for solving BCSP problems is that reducing the gap may require generating a large number of rth shortest paths. Desrochers and Soumis (1988) propose solving the BCSP problem by generalizing Dijkstra's shortest path algorithm. Dumitrescu and Boland (2003) use preprocessing techniques to accelerate the node-labeling algorithm, and demonstrate computationally that these methods can improve performance by an order of magnitude. Feng and Korkmaz (2015) and Pugliese and Gueriero (2013) provide some suggestions to the reduce the number of r^{th} shortest paths needed to close the gap. Feng and Korkmaz also discuss an extension of the *HZ* method to solve weight-constrained shortest path (WCSP) problems. Pugliese and Gueriero (2013) review the methodological literature for WCSP problems with multiple metrics, and even with negative arc costs.

Approximation Algorithms for the Diameter-Constrained and Hop-Constrained Minimum Spanning Tree (DCMST and HCMST) Problems Kortsarz and Peleg (1999) analyze the heuristic worst-case performance of a DCMST problem with maximum diameter of five. Marathe et al. (1998) develop approximation algorithms for a generalization of the Diameter-constrained Minimum Spanning Tree (DCMST) problem where the weights associated with arcs are not necessarily one. This generalization requires the total weight of the path connecting any pair of nodes in the tree to be less than or equal a specified value. The authors' approach starts with clusters consisting of single nodes, and sequentially merges these clusters until just one cluster remains, which is the heuristic solution. Hassin and Levin (2003) study a problem in which the diameter is not fixed, but rather pairs of nodes have hop limits that belong to $\{1, 2, \infty\}$. Assuming metric edge costs, they develop a constant ratio algorithm. They also consider cases where the graph induced by node-pairs with hop-limit of one or two is a Hamiltonian graph or a 2-vertex connected graph. Althaus et al. (2005) develop a randomized algorithm with approximation ratio of $O(log(n))$ for the HCMST problem with metric costs.

Polyhedral Results for Hop-Constrained Design Problems For the Hop-constrained Shortest Path (HCSP) problem, Dahl and Gouveia (2004) provide a complete characterization of the underlying polytope when the hop limit H is small. Dahl (1998) originally introduced the jump constraints while Grotschel and Stephan (2014) propose a systematic way of generating jump constraints as well as other inequalities via projection of a HCSP problem formulation that uses hop-indexed

variables (see Sect. 4.2.3). Although the basic jump inequalities do not necessarily define facets of the HCSP problem with general hop-limits, Reidl (2017) provides necessary and sufficient conditions for lifted jump inequalities to be facets of the HCSP polytope. Stephan (2009) identifies other facets by studying the polyhedral structure of related combinatorial problems.

Extended Formulations for Hop-Constrained Problems Gouveia (1998) is among the first researchers to study models with hop-indexed variables for hop-constrained problems. The variable disaggregation approach also extends to Hop-constrained Network Design (HCND) problems by defining hop-indexed flow variables for each commodity. This hop-indexed model is tighter than representing the hop constraints as routing requirements in formulation [*NDRR*], but the hop-indexed model does not fully close the integrality gap for HCND problems (unlike the situation for the HCSP problem). Researchers have used layered network representations and successfully applied the associated formulations with disaggregated (hop-indexed) flow variables to several special cases and variants of hop-constrained network design including minimum spanning tree problems with diameter constraints (e.g., Gouveia and Magnanti 2003; Gouveia et al. 2004, 2006), hub location with hop constraints (Camargo et al. 2017), and container shipping service selection with limited transshipments (Balakrishnan and Karsten 2017). The higher LP lower bounds of the hop-indexed model significantly accelerate branch-and-bound solution procedures for these problems.

Layered Network and Extended Formulations for Hop-Constrained Tree (HCT) Problems Gouveia et al. (2011), henceforth abbreviated as GSU, further strengthen this model by exploiting the problem's tree configuration requirement. Specifically, they show how to represent the HCT problem as a Steiner tree problem defined over a (single) layered network (instead of defining a separate layered network for each commodity). Effectively, this approach permits disaggregating the design variables (by hop index). For their equivalent Steiner tree problem, GSU consider a directed cut formulation (Maculan 1987) that contains only design variables (no flow variables since they do not consider routing costs), and uses cutset constraints to ensure connectivity from the root to every terminal node. GSU show that this model is tighter than the previous HCT formulation with hop-indexed flow variables defined over separate layered networks for each commodity. The authors also extend this approach to the DCMST problem. Their computational results demonstrate that using the stronger model reduces computational time by about two orders of magnitude compared to earlier methods (e.g., Gouveia and Magnanti 2003; Gouveia et al. 2004) that solve the model with only disaggregated flow variables. In Sect. 4.2.4, we discuss an extension of this technique of disaggregating the design variables to more general types of NDRR problems.

GSU's idea of disaggregating the design variables also relates to other interesting work. Ruthmair and Raidl (2011) apply disaggregation to Steiner tree problems with a single weight constraint for each commodity. To avoid solving the full extended formulation, they start with a small layered network obtained by deleting

some nodes, removing the outgoing arcs for each deleted node, and redirecting its incoming arcs to a corresponding node in an earlier (later) layer. Solving the reduced network provides a valid lower (upper) bound. The approach iteratively increases the size of the layered network until the upper and lower bounds are sufficiently close. Boland et al. (2017) developed and applied a similar strategy in the context of time-space networks (which are related to layered networks).

Another way of reducing the size of the layered network is to apply scaling-and-rounding (sometimes known as discretization) to the weight constraint to reduce the size of its coefficients. In the context of time-space networks, the approach of scaling-and-rounding the time unit does not seem to be as effective computationally as the methodology described above (Boland et al. 2017, 2019).

Extended Formulations for General NDRR Problems Researchers have recognized the benefits of reformulating various network design problems (e.g., facility location, Steiner trees, and uncapacitated fixed charge network design) using disaggregated variables and forcing constraints. For example, for fixed-charge network design, instead of using one commodity to represent all the flow originating from a source node, replacing the single commodity with multiple commodities, each corresponding to one destination served by that source, yields tighter model formulations and improved solution techniques (see Magnanti and Wong 1984). For research related to extended formulations, see, for example, Vanderbeck and Wolsey (2010), Conforti et al. (2010), Conforti et al. (2014), and Fiorini and Pashkovich (2015). Applying projection techniques (e.g., Conforti et al. 2014) to the extended formulations (with disaggregated flow or design variables) can yield valid inequalities and facets for the original problem formulation [*NDRR*]. For related work on projection techniques applied to the CSP polyhedron, see Coulard et al. (1994), and Grotschel and Stephan (2014). Mirchandani (2000) uses projection for the capacitated network loading problem, and Rardin and Wolsey (1993) use projection for uncapacitated network design models with multiple sources and sinks for each commodity. Gouveia et al. (2011) apply projection techniques to the single layered network model for HCT problems to obtain valid inequalities for the base model.

Column Generation Branch-and-price, which embeds the column generation technique within a branch-and-bound tree search framework (Barnhart et al. 1998; Desrosiers et al. 1995), has proved successful for various types of integer optimization problems. Column generation has been very successfully used in crew scheduling and routing problems, with routing requirements that reflect, for instance, time or distance limits of vehicle routes and work shifts (see Lubbecke and Desrosiers 2005 for an extensive list of references on applications of column generation).

Stabilization and other techniques can be useful in improving the convergence of column generation. Lubbecke and Desrosiers (2005) and Lemaréchal et al. (1995) discuss the use of a pre-chosen stabilization point. These general ideas have proven very successful in the special context of delay-constrained minimum spanning trees and Steiner trees. Computational tests by Leitner et al. (2012) show that the

stabilized version of column generation is about one order of magnitude faster than the usual column generation procedure for larger problems. The approach can solve difficult problems for networks with up to 999 nodes and about 10,000 arcs. Stabilized column generation is also at least competitive with state-of-the-art techniques using branch-and-cut applied to the single layered network model formulation (Gouveia et al. 2011).

Another convergence difficulty arises when the RMP primal solution is degenerate, implying multiple optimal dual prices in the master problem. Holloway (1973) suggests choosing an optimal dual solution that would benefit the overall convergence of column generation (i.e., choose a set of dual prices that would accurately reflect the optimal dual prices of the full problem). Such an approach would be similar in spirit to Magnanti and Wong (1981) (see also Rahmaniani et al. 2017) who propose exploiting degeneracy in Benders subproblems (whereas Holloway's proposal concerns degeneracy in the column generation master problem).

Uchoa et al. (2008) and Costa et al. (2019) discuss innovative branch-and-cut-and-price approaches to the Capacitated Minimum Spanning Tree (CMST) and the vehicle routing problems, respectively. Uchoa et al. use a new column space representation as well as robust cuts (see Poggi de Arago and Uchoa 2003) based on a new expanded representation of the arc flow variables based on capacity-indexed arc flows (similar to a layered network representation of arc flows) proposed by Gouveia and Martins (1999). Costa et al. (2019) give a comprehensive survey of techniques for improving the performance of column generation for vehicle routing including robust and non-robust cuts as well various other strategies. Some research has shown that the careful addition of non-robust cuts can improve the overall performance of the column generation approach. Thus, the cuts proposed in Sect. 3 (which are non-robust) might be of interest in the context of a column generation approach.

Benders Decomposition For an introduction to Benders decomposition and a general treatment of this approach, see, for example, Conforti et al. (2014). Over the years, researchers have suggested various techniques to improve Benders' classic approach. See Rahmaniani et al. (2017) for a comprehensive overview. As mentioned in Sect. 5.3, an active area of research is generating effective Benders feasibility cuts. See, for instance, Camargo et al. (2017).

Botton et al. (2013) apply Benders decomposition to a Hop-constrained Survivable Network Design problem. That is, in addition to hop constraints on the commodity routes, the problem also requires ensuring that each commodity has as least γ arc-disjoint origin-to-destination paths (which must all satisfy the hop limit restriction) in the chosen design. The objective is to minimize the total fixed cost of the chosen design arcs. For the design problem under consideration, at least one commodity's subproblem will always be infeasible until the relaxed master problem generates an optimal solution to the original design problem. The authors avoid this difficult issue by modifying the Benders branch-and-cut procedure by only occasionally generating Benders cuts (i.e. solving the subproblems). Thus, they trade off having a larger search tree for reducing the subproblem computation

time. This modified approach is an order of magnitude faster than the usual Benders decomposition approach. The decomposition approach can solve medium to large sized problems with up to 41 nodes and 820 edges and is significantly faster than solving the original formulation with CPLEX.

As mentioned in Sect. 5, strengthening the LP relaxation of the model formulation can improve the performance of Benders decomposition. Section 6.2 of Rahmaniani et al. (2017) discusses various research work on adding valid inequalities to the Benders master problem to strengthen the overall formulation. Magnanti and Wong (1981) discuss a framework for evaluating different model formulations (having different sets of subproblem variables) in the context of Benders decomposition based on their LP relaxation strength when the master problem variables have fixed values. Certain formulations can offer a richer (better) set of Benders cuts than other ones.

7 Concluding Remarks

In this chapter, we have reviewed various applications of network design with routing requirements, identified the challenges of solving this problem, and outlined modeling and solution approaches for the problem. We next summarize the key observations and learnings, and identify some opportunities for future work.

Applications Route constrained network design problems arise in a broad spectrum of industries. These include the transportation industry, where the NDRR problem applications comprise networks on the land, sea, and air. There are also many applications in telecommunications and other areas such as electricity distribution and machine scheduling.

Polyhedral Approach Using preprocessing techniques combined with insights about the problem structure, Balakrishnan et al. (2017) derive and implement computationally effective facets and valid inequalities for the NDRR problem. For the CNDRR problem, the added capacity constraints make solving the problem more challenging. So, using tight formulations, with added valid inequalities to strengthen the formulation, will be key for effective CNDRR solution performance. The ideas outlined in Sect. 3 to develop integrated inequalities that jointly consider the routing and capacity restrictions provide interesting and useful research directions to pursue.

Constrained Shortest Path heuristics There is a wide spectrum of creative approaches including Lagrangian relaxation with path enumeration, generalized Dijkstra algorithm with preprocessing, and scaling. Different goals (e.g., improving computational efficiency vs. improving worst-case bounds) result in different types of heuristic improvements. Can we obtain heuristic worst-case bounds for the weight-constrained shortest path problem with multiple routing requirements?

Worst-Case Analysis Sections 4.1.1 and 4.2.1 discuss the worst-case analysis of approximation algorithms for some special cases of the NDRR problem. These theoretical studies are challenging but facilitate our understanding about which particular problem characteristics make the NDRR problem easier or harder to solve. Can we build upon these previous studies to develop and analyze the worst-case performance of heuristics for more general NDRR problems?

Layered Networks The layered network approach uses variable disaggregation to obtain a tighter LP relaxation for hop-constrained spanning trees. The tighter formulation improves computational performance. Importantly, removing disaggregated variables in this model via projection constitutes a systematic method for obtaining polyhedral results in the original problem space (see previous discussion on extended formulations for general NDRR problems). Adopting and extending this approach for other problems appears to be promising.

The layered network variable disaggregation technique is different from previous approaches. Instead of disaggregating a commodity into a finer set of commodities (as researchers have previously done for facility location and uncapacitated fixed charge network design), it disaggregates a flow variable into a series of hop-indexed flow variables (or a design variable into a series of hop-indexed design variables). Could there be other new variable disaggregation schemes for different types of network design problems?

Decomposition Techniques Leveraging advances (over the past several decades) in decomposition techniques (e.g., stabilization, improved column pricing methods), embedding within a tree search procedure (e.g., branch-and-price or branch-and-cut) and exploiting the structural properties of the NDRR problem solution results in useful algorithms for some of its special cases. Further exploitation of these advances appears to be a promising area for future research. Moreover, decomposition techniques are more flexible and can address problem variants such as stochastic or prize-collecting variants more easily than other types of solution techniques.

Capacitated Network Design with Routing Restrictions Network design problems become more challenging to solve if we just add routing restrictions or arc capacity constraints. Even the simplest versions of the combined CNDRR model, which has both types of constraints are NP-hard, and are likely to be quite difficult to solve. Our discussion has highlighted the longitudinal (single commodity, multiple arc) structure versus lateral (single arc, multiple commodity) structure of the routing and capacity constraints. Developing effective solution methods will require leveraging and integrating the principles and approaches developed for the NDRR and capacitated network design models. For each of the modeling and methodological approaches (polyhedral methods, extended formulations, Lagrangian relaxation, column generation, and Benders decomposition) presented in this paper, we have also discussed possible ways to extend them to the CNDRR problem. Perhaps, research to solve the CNDRR problem can begin by first addressing its special cases such as the capacitated minimum spanning tree with hop limits, capacitated

hop-constrained network design, or two-commodity CNDRR before considering more general routing restrictions. These special cases have the advantage of providing a wider range of improved modeling options such as the layered network representation. Since there is little or no literature on the CNDRR problem and since this problem has considerable practical relevance, investigating and developing effective solution approaches for this problem is a promising and fruitful avenue for research.

References

Agarwal, R., & Ergun, O. (2008). Ship scheduling and network design for cargo routing in linear shipping. *Transportation Science, 42*, 175–196.

Ahuja, R. K., Jha, K. C., & Liu, J. (2007). Solving real-life railroad blocking problems. *Interfaces, 37*, 404–419.

Althaus, E., Funke, S., Har-Peled, S., Konemann, J., Ramos, E. A., & Skutella, M. (2005). Approximating k-hop minimum-spanning trees. *Operations Research Letters, 33*, 115–120.

Armacost, A. P., Barnhart, C., Ware, K. A. (2002). Composite variable formulations for express shipment service network design. *Transportation Science, 36*, 1–20.

Atamtürk, A., & Rajan, D. (2002). On splittable and unsplittable flow capacitated network design arc-set polyhedra . *Mathematical Programming, 92*, 315–333.

Balakrishnan A., & Altinkemer, K. (1992). Using a hop-constrained model to generate alternative communication network designs. *INFORMS Journal on Computing, 4*, 192–205.

Balakrishnan, A., & Karsten, C. V. (2017). Container shipping service selection and cargo routing with transshipment limits. *European Journal of Operational Research, 263*, 652–663.

Balakrishnan, A., Li, G., & Mirchandani, P. (2017). Optimal network design with end-to-end service requirements. *Operations Research, 65*, 729–750.

Balakrishnan, A., Magnanti, T. L., & Mirchandani, P. (1996) Heuristics, LPs, and trees on trees: Network design analyses. *Operations Research, 44*, 478–496.

Balakrishnan, A., Magnanti, T. L., & Mirchandani, P. (1997). Network design. In M. Dell'Amico, F. Maffioli, & S. Martello (Eds.), *Annotated bibliographies in combinatorial optimization* (pp. 311–334). New York: John Wiley and Sons.

Balakrishnan, A., Magnanti, T. L., & Wong, R. T. (1989). A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research, 37*, 716–740.

Balakrishnan, A., Mirchandani, P., & Natarajan, H. P. (2009). Connectivity upgrade models for survivable network design. *Operations Research, 57*, 170–186.

Balakrishnan, A., Mirchandani, P., & Wong, R. T. (2020). On multi-constrained path, tree, and network design problems. Working paper

Barnhart, C., Jin, H., & Vance, P. (2000). Railroad blocking: A network design applications. *Operations Research, 48*, 603–614.

Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M., & Vance, P. (1998) Branch-and-price: Column generation for solving huge integer programs. *Operations Research, 46*, 316–329.

Barnhart, C., & Schneur, R. (1996). Air network design for express shipment service . *Operations Research, 44*, 852–863.

Benhamiche, A., Mahjoub, A. R., Perrot, N., & Uchoa, E. (2016). Unsplittable non-additive capacitated network design using set functions polyhedra. *Computers and Operations Research, 66*, 105–115.

Bienstock, D., Günlük, O. (1996). Capacitated network design – Polyhedral structure and computation. *INFORMS Journal on Computing, 8*, 243–259.

Boland, N., Hewitt, M., Marshall, L., & Savelsbergh, M. (2017). The continuous-time service network design problem. *Operations Research, 65*, 1303–1321.

Boland N, Hewitt M, Marshall, L., & Savelsbergh, M. (2019). The price of discretizing time: a study in service network design. *Euro Journal on Transportation and Logistics, 8*, 195–216.

Botton, Q., Fortz, B., Gouveia, L., & Poss, M. (2013). Benders decomposition for the hop-constrained survivable network design problem. *INFORMS Journal on Computing, 25*, 13–26.

Brockmüller, B., Günlük, O., & Wolsey, L. A. (2004). Designing private line networks: polyhedral analysis and computation. *Transactions on Operational Research, 16*, 7–24.

Camargo, R., de Miranda, G., Jr., O'Kelly, M., & Campbell, J. (2017). Formulations and decomposition methods for the incomplete hub location problem with and without hop-constraints. *Applied Mathematical Modelling, 51*, 274–301.

Conforti, M., Cornuejols, G., & Zambelli, G. (2010). Extended formulations in combinatorial optimization. *4OR: A Quarterly Journal of Operations Research, 8*, 1–48.

Conforti, M., Cornuejols, G., & Zambelli, G. (2014). *Integer programming*. Heidelberg, Springer.

Costa, L., Contardo, C., & Desaulniers, G. (2019). Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science, 53*, 946–985.

Coulard, C., Gamble, B., & Liu, J. (1994). The K-walk polyhedron. In D.-Z. Du & J. Sen (Eds.), *Advances in optimization and approximation*. Dordrecht: Kluwer Academic Publishers.

Crainic, T. G. (2000). Service network design in freight transportation. *European Journal of Operational Research, 122*, 272–288.

Dahl, G. (1998). The 2-hop spanning tree problem. *Operations Research Letters, 23*, 21–26.

Dahl, G., & Gouveia, L. (2004). On the directed hop-constrained shortest path problem. *Operations Research Letters, 32*, 15–22.

De Boeck, J., & Fortz, B. (2017). Extended formulation for hop constrained distribution network configuration problems. *European Journal of Operational Research, 265*, 488–502.

Desaulniers, G., Madsen, O. B., & Ropke, S. (2014). The vehicle routing problem with time windows. In P. Toth, & D. Vigo (Eds.), *Vehicle routing: Problems, methods, and applications, MOS-SIAM series on optimization* (Vol. 18, pp. 119–159). Philadelphia: SIAM.

Desrochers, M., & Soumis, F. (1988). A generalized permanent labelling algorithm for the shortest path problem with time windows. *INFOR: Information Systems and Operational Research, 26*, 191–212.

Desrosiers, J., Dumas, Y., Solomon, M., & Soumis, F. (1995). Time constrained routing and scheduling. In M. Ball, T. L. Magnanti, C. Monma, & G. L. Nemhauser (Eds.), *Handbooks in operations research and management science* (Vol. 8, pp. 35–139). Amsterdam: Elsevier.

Dumitrescu, I., & Boland, N. (2003). Improved preprocessing, labeling, and scaling algorithms for the weight-constrained shortest path problem. *Networks, 42*, 135–153.

Ergun, F., Sinha, R., & Zhang, L. (2002). An improved FPTAS for restricted shortest path. *Information Processing Letters, 83*, 287–291.

Estrada, M., & Robuste, F. (2009). Long-Haul shipment optimization for less-than-truckload carriers. *Transportation Research Record: Journal of the Transportation Research Board, 2091*, 12–20.

Feng, G., & Korkmaz, T. (2015). Finding multi-constrained multiple shortest paths. *IEEE Transactions on Computers, 64*, 2559–2572.

Fiorini, S., & Pashkovich, K. (2015). Uncapacitated flow-based extended formulations. *Mathematical Programming, 153*, 117–131.

Garey, M. R., & Johnson, D. S. (2002). *Computers and intractability: A guide to the theory of NP completeness*. San Francisco: W. H. Freeman.

Gendron, B., Crainic, T.G., & Frangioni, A. (1999). Multicommodity capacitated network design. In B. Sansò & P. Soriano (Eds.), *Telecommunications network planning* (pp. 1–19). Centre for Research on Transportation. Boston: Springer.

Gopalakrishnan, B., & Johnson, E. L. (2005). Airline crew scheduling: State-of-the-art. *Annals of Operations Research 140*, 305–337.

Gouveia, L. (1998). Using variable redefinition for computing lower bounds for minimum spanning tree and Steiner tree with hop constraints. *INFORMS Journal on Computing, 10*, 180–188.

Gouveia, L., & Magnanti, T. L. (2003). Network flow models for designing diameter-constrained spanning and Steiner trees. *Networks, 41*, 159–173.

Gouveia, L., Magnanti, T. L., & Requejo, C. (2004). A 2-path approach for odd diameter-constrained minimum spanning and Steiner trees. *Networks, 44*, 254–265.

Gouveia, L., Magnanti, T. L., & Requejo, C. (2006) An intersecting tree model for odd-iameter-constrained minimum spanning and Steiner trees. *Annals of Operations Research, 146*, 19–39.

Gouveia, L., & Martins, P. (1999). The capacitated minimal spanning tree problem: An experiment with a hop-indexed model. *Annals of Operations Research 86*, 271–294.

Gouveia, L., Simonetti, L., & Uchoa, E. (2011). Modeling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs. *Mathematical Programming, 128*, 123–148.

Grandoni, F., Ravi, R., Singh, M., & Zenklusen, R. (2014). New approaches to multi-objective optimization. *Mathematical Programming, 146*, 525–554.

Grötschel, M., Monma, C. L., & Stoer, M. (1995). Design of survivable networks. In M. Ball, T. L. Magnanti, C. Monma, G. L. Nemhauser (Eds.), *Handbooks in operations research and management science* (Vol. 7, pp. 617–672). Amsterdam: Elsevier.

Grötschel, M., & Stephan, R. (2014). Characterization of facets of the hop-constrained chain polytope via dynamic programming. *Discrete Applied Mathematics, 162*, 229–246.

Handler, G., & Zang, I. (1980). A dual algorithm for the constrained shortest path problem. *Networks, 10*, 293–309.

Hassin, R. (1992). Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research, 17*, 36–42.

Hassin, R., & Levin, A. (2003). Minimum spanning tree with hop restrictions. *Journal of Algorithms, 48*, 220–238.

Holloway, C. (1973). A generalized approach to Dantzig-Wolfe decomposition for concave programs. *Operations Research, 21*, 210–220.

Karsten, C. V., Brouer, B. D., Desaulniers, G., & Pisinger, D. (2017). Time constrained liner shipping network design. *Transportation Research Part E, 105*, 152–162.

Kortsarz, G., & Peleg, D. (1999). Approximating the weight of shallow Steiner trees. *Discrete Applied Mathematics, 93*, 265–285.

Lawler, E. L. (1976). *Combinatorial optimization: Networks and matroids*. New York: Courier Corporation.

Leitner, M., Ruthmair, M., & Raidl, G. (2012). Stabilizing branch-and-price for constrained tree problems. *Networks, 61*, 150–170.

Lemaréchal, C., Nemirovskii, A., & Nesterov, Y. (1995). New variants of bundle methods. *Mathematical Programming, 69*, 111–147.

Lorenz, D. H., & Raz, D. (2001). A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters, 28*, 213–219.

Lübbecke, M. E., & Desrosiers, J. (2005). Selected topics in column generation. *Operations Research, 53*, 1007–1023.

Maculan, N. (1987). The Steiner problem in graphs. *Annals of Discrete Mathematics, 31*, 185–212.

Magnanti, T. L., Mirchandani, P., & Vachani, R. (1993). The convex hull of two core capacitated network design problems. *Mathematical Programming, 60*, 233–250.

Magnanti, T. L., Mirchandani, P., & Vachani, R. (1995). Modeling and solving the two-facility capacitated network loading problem. *Operations Research, 43*, 142–157.

Magnanti, T. L., & Wong, R. T. (1981). Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research, 29*, 464–484.

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

Malandraki, C., Zaret, D., Perez, J., & Holland, C. (2001). Industrial engineering applications in transportation. In G. Salvendy (Eds.), *Handbook of industrial engineering* (3rd ed., pp. 787–824). New York: John Wiley and Sons.

Marathe, M. V., Ravi, R., Sundaram, R., Ravi, S. S., Rosenkrantz, D. J., & Hunt, H. B. (1998). Bicriteria network design problems. *Journal of Algorithms, 28*, 142–171.

Mirchandani, P. (2000). Projections of the capacitated network loading problem. *European Journal of Operational Research, 122*, 534–560.

Nemhauser, G. L., & Wolsey, L. A. (1988). *Integer and combinatorial optimization*. New York: Wiley.

Poggi de Arago, M., & Uchoa, E. (2003). Integer program reformulation for robust branch-and-cut-and-price. In L. Wolsey (Ed.), *Annals of Mathematical Programming in Rio* (pp. 59–61)

Pugliese, L. D. P., & Guerriero, F. (2013). A survey of resource constrained shortest path problems: Exact solution approaches. *Networks, 62*, 183–200.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research, 259*, 801–817.

Rardin, R. L., & Choe, U. (1979). *Tighter relaxations of fixed charge network flow problems*. Industrial and Systems Engineering Report J-79-18, Georgia Institute of Technology

Rardin, R. L., & Wolsey, L. A. (1993). Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems. *European Journal of Operational Research, 71*, 95–109.

Reidl, W. (2017). A complete characterization of jump inequalities for the hop-constrained shortest path problem. *Discrete Applied Mathematics, 225*, 85–113.

Ruthmair, M., & Raidl, G. (2011). Layered graph model and an adaptive layers framework to solve delay–constrained minimum tree problems. In O. Gunluk & G. Woeginger (Eds.), *IPCO 2011* (pp. 276–288). Berlin Heidelberg, Springer-Verlag.

Stephan, R. (2009). Facets of the (s,t)-path polytope. *Discrete Applied Mathematics, 157*, 3119–3132.

Uchoa, E., Fukasawa, F., Lysgaard, J., Pessoa, A., Poggi de Arago, M., & Andrade, D. (2008). Robust branch-cut-and-price for the capacitated minimum spanning tree problem over a large extended formulation. *Mathematical Programming, 112*, 443–472.

Vanderbeck, F., & Wolsey, L. A. (2010). Reformulation and decomposition of integer programs. In M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, & L. A. Wolsey (Eds.), *50 Years of integer programming 1958–2008* (pp. 431–502). Berlin Heidelberg: Springer-Verlag.

Vazirani, V. V. (2013). *Approximation algorithms*. New York: Springer Science & Business Media.

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research, 231*:1–21.

Warburton, A. (1987). Approximation of pareto optima in multiple–objective, shortest-path problems. *Operations Research, 35*, 70–79.

Wilhelm, W. E., Damodaran, P., & Li, J. (2003). Prescribing the content and timing of product upgrades. *IIE Transactions, 35*, 647–663.

Williamson, D. P., & Shmoys, D. B. (2011). *The design of approximation algorithms*. New York: Cambridge University Press.

Wolsey, L. (2011). Using extended formulations in practice. *Optima, 85*, 7–9.

Yildiz, B., & Savelsbergh, M. (2019). Optimizing package express operations in China. Optimization Online 6799.

Zabarankin, M., Uryasev, S., & Pardalos, P. (2001). Optimal risk path algorithms. In R. Murphey & P. Pardalos (Eds.), *Cooperative control and optimization* (pp. 271–303). Dordrecht: Kluwer.

Zhu, E., Crainic, T. G., & Gendreau, M. (2014). Scheduled service network design for freight rail transportation. *Operations Research, 62*, 383–400.