Springer Optimization and Engineering manuscript No.
(will be inserted by the editor)

Distributed Robust Optimization (DRO)
Part I: Framework and Example

**Kai Yang · Jianwei Huang · Yihong Wu ·
Xiaodong Wang · Mung Chiang**

the date of receipt and acceptance should be inserted later

Abstract Robustness of optimization models for network problems in communication networks has been an under-explored topic. Most existing algorithms for solving robust optimization problems are centralized, thus not suitable for networking problems that demand distributed solutions. This paper represents a first step towards a systematic theory for designing distributed *and* robust optimization models and algorithms. We first discuss several models for describing parameter uncertainty sets that can lead to decomposable problem structures and thus distributed solutions. These models include ellipsoid, polyhedron, and D-norm uncertainty sets. We then apply these models in solving a robust rate control problem in wireline networks. Three-way tradeoffs among performance, robustness, and distributiveness are illustrated both analytically and through simulations. In Part II of this two-part paper, we will present applications to wireless power control using the framework of distributed robust optimization.

1 Introduction

Despite the importance and success of using optimization theory to study communication and network problems, most work in this area makes the unrealistic assump-

Kai Yang
Bell Labs, Alcatel-Lucent, Murray Hill, NJ, USA
E-mail: kai.yang@alcatel-lucent.com

Jianwei Huang
Department of Information Engineering, The Chinese University of Hong Kong, China
E-mail: jwhuang@ie.cuhk.edu.hk

Yihong Wu
Department of Electrical Engineering, Princeton University, Princeton, NJ, USA
E-mail: yihongwu@princeton.edu

Xiaodong Wang
Department of Electrical Engineering, Columbia University, New York, NY, USA
E-mail: wangx@ee.columbia.edu

Mung Chiang
Department of Electrical Engineering, Princeton University, Princeton, NJ, USA
E-mail: chiangm@princeton.edu

tion that the data defining the constraints and objective function of the optimization problem can be obtained precisely. We call the corresponding problems "nominal". In many practical problems, these data are typically inaccurate, uncertain, or time-varying. Solving the nominal optimization problems may lead to poor or even infeasible solutions when deployed.

Over the last ten years, robust optimization has emerged as a framework of tackling optimization problems under data uncertainty (e.g., [1,2,3,4,5]). The basic idea of robust optimization is to seek a solution which remains feasible and near-optimal under the perturbation of parameters in the optimization problem. Each robust optimization problem is defined by three-tuple: *a nominal formulation, a definition of robustness*, and *a representation of the uncertainty set*. The process of making an optimization formulation robust can be viewed as a mapping from one optimization problem to another. A central question is as follows: when will important properties, such as convexity and decomposability, be preserved under such mapping? In particular, what kind of nominal formulation and uncertainty set representation will preserve convexity and decomposability in the robust version of the optimization problem?

So far, almost all of the work on robust optimization focuses on determining what representations of data uncertainty preserves convexity, thus tractability through a centralized solution, in the robust counter part of the nominal problem for a given definition of robustness. For example, for the worst-case robustness, it has been shown that under the assumption of ellipsoid set of data uncertainty, a robust linear optimization problem can be converted into a second-order cone problem; and a robust second-order cone problem can be reformulated as a semi-definite optimization problem [6]. The reformulations can be solved efficiently using centralized algorithms such as the interior-point method.

In this paper, motivated by needs in communication networking, we will focus instead on the *distributiveness*-preserving formulation of the robust optimization. The driving question thus becomes: how much more communication overhead is introduced in making the problem robust?

To develop a systematic theory of Distributed Robust Optimization (DRO), we first show how to represent an uncertainty set in a way that not only captures the data uncertainty in the model but also leads to a distributively solvable optimization problem. Second, in the case where a fully distributed algorithm is not obtainable, we focus on the *tradeoff* between robustness and distributiveness. Distributed algorithms are often developed based on decomposability structure of the problem, which may disappear as the optimization formulation is made robust. While distributed computation has long been studied [7], unlike convexity of a problem, distributiveness of an algorithm does not have a widely-agreed definition. It is often quantified by the amount of communication overhead required: how far and how frequent do the nodes have to pass message around? Zero communication overhead is obviously the "most distributed", and we will see how the amount of overhead trades-off with the degree of robustness.

In Section 2, we develop the framework of DRO, with a focus on the characterization of uncertainty sets that are useful for designing distributed algorithms. An example on robust rate control is given in Section 3, where we discuss various tradeoffs among robustness, distributiveness, and performance through both analysis and numerical studies. Conclusions to this part are given in Section 4. In Part II of the paper, extensive applications of DRO to wireless power control will be presented

2 General Framework

2.1 Optimization Models

There are many uncertainties in the design of communication networks. These uncertainties stem from various sources and can be broadly grouped into two categories. The first type of uncertainties are related to the perturbation of a set of design parameters due to erroneous inputs such as errors in estimation or implementation. We call them *perturbation errors*. For example, power control is often used to maintain the quality of a wireless link, i.e., the signal-to-interference-plus-noise ratio (SINR). However, many power control schemes are optimized for estimated parameters, e.g., wireless channel quality or interference from other users. In reality, precise information on these parameters are rarely available. Consequently, the corresponding optimized power control scheme could easily become infeasible or exhibit poor performance. A good power control scheme should be capable of protecting the link quality against the perturbation errors. We show such an example on robust wireless power control in the second part of this paper. A common characteristic of such perturbation errors is that they can often be modeled as a *continuous* uncertainty set surrounding the basic point estimate of the parameter. The size of the uncertainty set could be used to characterize the level of perturbations the designer needs to protect against. The second type of uncertainties is termed as *disruptive errors*, as they are caused by the failure of communication links within the network. This type of errors can be modeled as a *discrete* uncertainty set. In Section III of this paper, we present an example related to disruptive errors on robust multipath rate control for service reliability.

To make our discussions concrete, we will focus on a class of optimization problems with the following nominal form: maximization of a *concave* objective function over a given data set characterized by *linear* constraints,

$$\begin{aligned} &\text{maximize} \;\; f_0\left(\boldsymbol{x}\right) \\ &\text{subject to} \;\; \boldsymbol{A}\boldsymbol{x} \preceq \boldsymbol{b} \\ &\text{variables} \;\; \boldsymbol{x}, \end{aligned} \tag{1}$$

where $\boldsymbol{A}$ is an $M \times N$ matrix, $\boldsymbol{x}$ is an $N \times 1$ vector, and $\boldsymbol{b}$ is an $M \times 1$ vector. We use $\preceq$ to denote component-wise inequality. This class of problems can model a wide range of engineering systems (e.g., [8,9,10,11,12]). Generalization to nonlinear and convex constraint sets presents a direction for major future work.

The uncertainty of Problem (1) may exist in the objective function f_0, matrix parameter $\boldsymbol{A}$, and vector parameter $\boldsymbol{b}$. In many cases, the uncertainty in objective function f_0 can be converted into uncertainty of the parameters defining the constraints [13]. Later in Section 3 we show that it is also possible to convert the uncertainty in $\boldsymbol{b}$ into uncertainty in $\boldsymbol{A}$ in certain cases (although this could be difficult in general). In the rest of the paper, we will focus on studying the uncertainty in $\boldsymbol{A}$. In many networking problems, structures and physical meaning of matrix $\boldsymbol{A}$ readily leads to distributed algorithms, e.g., in rate control where $\boldsymbol{A}$ is a given routing matrix, distributed subgradient algorithm is well-known to solve the problem, with an interesting correspondence with the practical protocol of TCP. Making the optimization robust may turn the linear constraints into nonlinear ones and increase the amount of message passing. Quantifying the tradeoff between robustness and distributedness is a main subject to study in this paper.

In the robust counterpart of Problem (1), we require the constraints $\boldsymbol{A}\boldsymbol{x} \preceq \boldsymbol{b}$ to be valid for any $\boldsymbol{A} \in \mathcal{A}$, where $\mathcal{A}$ denotes the uncertainty set of $\boldsymbol{A}$, and the definition of robustness is in the *worst-case* sense [14]. Notice that we assume the uncertainty set $\mathcal{A}$ is either an accurate description or a conservative estimation of the uncertainty in practice. In other words, it is not possible to have parameters outside set $\mathcal{A}$. In this case, the definition of robustness is the *worst case robustness*, i.e., the solution of the robust optimization problem is always feasible. However, this approach might be too conservative. A more meaningful choice of robustness is the *chance-constrained robustness*, i.e., the probability of infeasibility (or outage) is upper bounded. We can flexibly adjust the chance-constrained robustness of the robust solution by solving the worst-case robust optimization problem over a properly selected subset of the exact uncertainty set. We will discuss such an example in details in Section 3.4.

If we allow an arbitrary uncertainty set $\boldsymbol{A}$, then the robust optimization problem is difficult to solve even in a centralized manner [15]. In this paper, we will focus on the study of *constraint-wise* (i.e. *row-wise*) uncertainty set, where the uncertainties between different rows in matrix $\boldsymbol{A}$ are decoupled. This restricted class of uncertainty set characterizes the data uncertainty in many practical problems, and it also allows us to convert the robust optimization problem into a formulation that is distributively solvable. Tackling more general forms of uncertainties is another direction of extension in the future.

Denote the j^{th} row of $\boldsymbol{A}$ be $\boldsymbol{a}_j^T$, which lies in a compact uncertainty set $\mathcal{A}_j$. Then the *robust* optimization problem that we focus on in this paper can be written in the following form:

$$\begin{aligned} &\text{maximize } \; f_0(\boldsymbol{x}), \\ &\text{subject to } \; \boldsymbol{a}_j^T \boldsymbol{x} \le b_j, \; \forall \boldsymbol{a}_j \in \mathcal{A}_j, \; \forall 1 \le j \le M, \\ &\text{variables } \; \boldsymbol{x}. \end{aligned} \tag{2}$$

We show that the robust optimization problem (2) can be equivalently written in a form represented by *protection functions* instead of uncertainty sets. Denote the nominal counterpart of problem (2) with a coefficient matrix $\bar{\boldsymbol{A}}$ (i.e., the values when there is no uncertainty), with the j^{th} row's coefficient $\bar{\boldsymbol{a}}_j \in \mathcal{A}_j$. Then

Proposition 1 *Problem (2) is equivalent to the following* convex *optimization problem:*

$$\begin{aligned} &\textit{maximize } \; f_0(\boldsymbol{x}), \\ &\textit{subject to } \; \bar{\boldsymbol{a}}_j^T \boldsymbol{x} + g_j(\boldsymbol{x}) \; \le b_j, \; \forall 1 \le j \le M, \\ &\textit{variables } \; \boldsymbol{x}, \end{aligned} \tag{3}$$

where

$$g_j(\boldsymbol{x}) = \max_{\boldsymbol{a}_j \in \mathcal{A}_j} (\boldsymbol{a}_j - \bar{\boldsymbol{a}}_j)^T \boldsymbol{x}, \tag{4}$$

is the protection function for the j^{th} constraint, which depends on the uncertainty set $\mathcal{A}_j$ and the nominal row $\bar{\boldsymbol{a}}_j$. Furthermore, $g_j(\boldsymbol{x})$ is a convex function since for any $0 \le t \le 1$, we have that

$$\max_{\boldsymbol{a}_j \in \mathcal{A}_j} (\boldsymbol{a}_j - \bar{\boldsymbol{a}}_j)^T [t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_2] \le t \max_{\boldsymbol{a}_j \in \mathcal{A}_j} (\boldsymbol{a}_j - \bar{\boldsymbol{a}}_j)^T \boldsymbol{x}_1 + (1-t) \max_{\boldsymbol{a}_j \in \mathcal{A}_j} (\boldsymbol{a}_j - \bar{\boldsymbol{a}}_j)^T \boldsymbol{x}_2. \tag{5}$$

Different forms of $\mathcal{A}_j$ will lead to different protection function $g_j(\boldsymbol{x})$, which results in different robustness and performance tradeoff of the formulation. Next we consider several approaches in terms of modeling $\mathcal{A}_j$ and the corresponding protection function $g_j(\boldsymbol{x})$.

2.2 Robust Formulation Defined By Polyhedron Uncertain Set

In this case, the uncertainty set $\mathcal{A}_j$ is a polyhedron characterized by a set of linear inequalities, i.e., $\mathcal{A}_j \triangleq \{\boldsymbol{a}_j : \boldsymbol{D}_j \boldsymbol{a}_j \preceq \boldsymbol{c}_j\}$. The protection function is

$$g_j(\boldsymbol{x}) = \max_{\boldsymbol{a}_j : \boldsymbol{D}_j \boldsymbol{a}_j \preceq \boldsymbol{c}_j} (\boldsymbol{a}_j - \bar{\boldsymbol{a}}_j)^T \boldsymbol{x}, \tag{6}$$

which involves a linear program (LP). We next show that the uncertainty set can be translated into a set of linear constraints. In the j^{th} constraint in (2), with $\boldsymbol{x} = \hat{\boldsymbol{x}}$ fixed, we can characterize the set $\forall \boldsymbol{a}_j \in \mathcal{A}_j$ by comparing b_j with the outcome of the following LP:

$$v_j^* = \max_{\boldsymbol{a}_j : \boldsymbol{D}_j \boldsymbol{a}_j \preceq \boldsymbol{c}_j} \boldsymbol{a}_j^T \hat{\boldsymbol{x}}. \tag{7}$$

If $v_j^* \leq b_j$, then $\hat{\boldsymbol{x}}$ is feasible for (2). However, this approach is not very useful since it requires solving one LP in (7) for each possible $\hat{\boldsymbol{x}}$. Alternatively, we take the Lagrange dual problem of the LP in (7),

$$v_j^* = \min_{\boldsymbol{p}_j : \boldsymbol{D}_j^T \boldsymbol{p}_j \succeq \hat{\boldsymbol{x}}, \boldsymbol{p}_j \succeq \boldsymbol{0}} \boldsymbol{c}_j^T \boldsymbol{p}_j. \tag{8}$$

If we can find a feasible solution $\hat{\boldsymbol{p}}_j$ for (8), and $\boldsymbol{c}_j^T \hat{\boldsymbol{p}}_j \leq b_j$, then we must have $v_j^* \leq \boldsymbol{c}_j^T \hat{\boldsymbol{p}}_j \leq b_j$. We can thus replace constraints in (2) by the following constraints:

$$\boldsymbol{c}_j^T \boldsymbol{p}_j \leq b_j, \ \boldsymbol{D}_j^T \boldsymbol{p}_j \succeq \boldsymbol{x}, \ \boldsymbol{p}_j \succeq \boldsymbol{0}, \ \forall 1 \leq j \leq M, \tag{9}$$

and we now have an equivalent and *deterministic* formulation for Problem (2), where all the constraints are linear.

2.3 Robust Formulation Defined by D-norm Uncertainty Set

D-norm approach [13] is another method to model the uncertainty set, and has advantages such as guarantee of feasibility independent of uncertainty distributions and flexibility in trading off between robustness and performance.

Consider the j^{th} constraint $\boldsymbol{a}_j^T \boldsymbol{x} \leq b_j$ in (2). Denote the set of all uncertain coefficients in $\boldsymbol{a}_j$ as $\mathcal{E}_j$. The size of $\mathcal{E}_j$ is $|\mathcal{E}_j|$, which might be smaller than the total number of coefficients N (i.e., a_{ij} for some i might not have uncertainty). For each $a_{ij} \in \mathcal{E}_j$, assume the actual value falls into the range of $[\bar{a}_{ij} - \hat{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}]$, in which $\hat{a}_{ij}$ is a given error bound. Also choose a nonnegative integer $\Gamma_j \leq |\mathcal{E}_j|$. The definition of robustness associated with the D-norm formulation is to maintain feasibility if at most Γ_j out of all possible $|\mathcal{E}_j|$ parameters are perturbed. Let's denote $\mathcal{S}_i$ as the set of

Γ_j uncertain coefficients. The above robustness definition can be characterized by the following protection function,

$$g_j(\boldsymbol{x}) = \max_{\mathcal{S}_j:\mathcal{S}_j\subseteq\mathcal{E}_j,|\mathcal{S}_j|=\Gamma_j} \sum_{i\in\mathcal{S}_j} \hat{a}_{ij}|x_i|. \tag{10}$$

If $\Gamma_j = 0$, then $g_j(\Gamma_j, \boldsymbol{x}) = 0$ and the j^{th} constraint is reduced to the nominal constraint. If $\Gamma_j = |\mathcal{E}_j|$, then $g_j(\boldsymbol{x}) = \sum_{i\in\mathcal{E}_j} \hat{a}_{ij}|x_i|$ and the j^{th} constraint becomes Soyster's worst-case formulation [13]. The tradeoff between robustness and performance can be obtained by adjusting Γ_j.

Note that the nonlinearity of $g_j(\Gamma_j, \boldsymbol{x})$ is difficult to deal with in the constraint. We can formulate it into the following linear integer programming problem,

$$\max_{\{s_{ij}\in\{0,1\}\}\forall i\in\mathcal{E}_j} \sum_{i\in\mathcal{E}_j} \hat{a}_{ij}|x_i|s_{ij}, \text{ s.t. } \sum_{i\in\mathcal{E}_j} s_{ij} \leq \Gamma_j. \tag{11}$$

Thanks to its special structure, (11) has the same optimal objective function value as the following linear programming problem,

$$\max_{\{0\leq s_{ij}\leq 1\}\forall i\in\mathcal{E}_j} \sum_{i\in\mathcal{E}_j} \hat{a}_{ij}|x_i|s_{ij}, \text{ s.t. } \sum_{i\in\mathcal{E}_j} s_{ij} \leq \Gamma_j. \tag{12}$$

Taking the dual of Problem (12), we have

$$\min_{\{p_{ij}\geq 0\}\forall i\in\mathcal{E}_j, q_j\geq 0} q_j\Gamma_j + \sum_{i\in\mathcal{E}_j} p_{ij}, \text{ s.t. } q_j + p_{ij} \geq \hat{a}_{ij}|x_i|, \forall i \in \mathcal{E}_j. \tag{13}$$

Similar to Section 2.2, we can substitute (13) into the robust Problem (2) to obtain an equivalent formulation:

$$\begin{aligned} &\text{maximize} \;\; f_0(\boldsymbol{x}) \\ &\text{subject to} \;\; \sum_i \bar{a}_{ij}x_i + q_j\Gamma_j + \sum_{i\in\mathcal{E}_j} p_{ij} \leq b_j, \forall j, \\ &\qquad\qquad q_j + p_{ij} \geq \hat{a}_{ij}y_i, \;\; \forall i \in \mathcal{E}_j, \forall j, \\ &\qquad\qquad -y_i \leq x_i \leq y_i, \forall i, \\ &\text{variables} \;\; \boldsymbol{x}, \; \boldsymbol{y} \succeq \boldsymbol{0}, \; \boldsymbol{p} \succeq \boldsymbol{0}, \; \boldsymbol{q} \succeq \boldsymbol{0}. \end{aligned} \tag{14}$$

2.4 Robust Formulation Defined by Ellipsoid Uncertainty Set

Ellipsoid is commonly used to approximate complicated uncertainty sets for statistical reasons [15] and for its ability to succinctly describe a set of discrete points in Euclidean geometry [14]. Here we consider the case where coefficient $\boldsymbol{a}_j$ falls in an ellipsoid centered at the nominal $\bar{\boldsymbol{a}}_j$. Specifically,

$$\mathcal{A}_j = \{\bar{\boldsymbol{a}}_j + \Delta\boldsymbol{a}_j : \; \sum_i |\Delta a_{ij}|^2 \leq \epsilon_j^2\}. \tag{15}$$

By (4), the protection function is given by

$$g_j(\boldsymbol{x}) = \max\left\{\sum_i \Delta a_{ij}x_i : \sum_i |\Delta a_{ij}|^2 \leq \epsilon_j^2\right\}, \tag{16}$$

Denote by $\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$ as the ℓ_2-norm (or the Euclidean norm) of $\boldsymbol{x}$. By Cauchy-Schwartz inequality,

$$\sum_i \Delta a_{ij} x_i \leq \sqrt{\sum_i |\Delta a_{ij}|^2} \|\boldsymbol{x}\|_2 \leq \epsilon_j \|\boldsymbol{x}\|_2,$$

and the equality is attained by choosing

$$\Delta a_{ij} = \frac{x_i \epsilon_j}{\|\boldsymbol{x}\|_2}.$$

Therefore we conclude that

$$g_j(\boldsymbol{x}) = \epsilon_j \|\boldsymbol{x}\|_2. \tag{17}$$

Although the resulting constraint in Problem (2) is not readily decomposable using standard decomposition techniques, we will show in Part II of this paper that this leads to tractable formulations in some important applications (e.g., wireless power control) where users can obtain network information through local measurements without global message passing.

2.5 Distributed Algorithm for Robust Optimization under Linear Constraints

We now consider possible distributed algorithms to solve the general robust optimization problem with linear constraints in (2). Notice, however, the design of a truly distributed algorithm needs to take into account the setup of the practical system. Our motivation here is not to design a general distributed algorithm which is guaranteed to work equally well for all applications. Instead, we try to show how to solve (2) by exploring its special structures in DRO, and the resulting algorithm may lead to distributed algorithm for each of the individual applications, such as those in Section III below and Part II of the paper.

Primal-dual Cutting-plane Method: The cutting-plane method has been used for solving the general nonlinear programming problems and mixed linear integer programming problems [16]. We assume the protection function here is convex and bounded as in the previous subsections. Also, we assume the feasible region of the robust optimization, X, is a convex and bounded set. At the k^{th} iteration of this method, we use the following function to approximate $g_j(\boldsymbol{x})$,

$$\bar{g}_j(\boldsymbol{x}) = \max_{\ell} \quad \tilde{\boldsymbol{a}}_{j\ell}^T \boldsymbol{x} + \tilde{b}_{j\ell}, \ 1 \leq \ell \leq k, \tag{18}$$

such that $\forall \boldsymbol{x} \in X$, we have

$$\bar{g}_j(\boldsymbol{x}) \leq g_j(\boldsymbol{x}). \tag{19}$$

The key is to appropriately choose $\tilde{\boldsymbol{a}}_{j\ell}$ and $\tilde{b}_{j\ell}$ for all ℓ, i.e., generating the proper cutting planes.

Substituting the protection function $g_j(\boldsymbol{x})$ with $\bar{g}_j(\boldsymbol{x})$, we obtain the following approximating problem.

$$\begin{aligned} &\text{maximize} \ \ f_0(\boldsymbol{x}), \\ &\text{subject to} \ \ \bar{\boldsymbol{a}}_j^T \boldsymbol{x} + \hat{\boldsymbol{a}}_{j\ell}^T \boldsymbol{x} + \tilde{b}_{j\ell} \ \leq b_j, \ \forall 1 \leq j \leq M, \ 1 \leq \ell \leq k, \\ &\text{variables} \ \ \boldsymbol{x}. \end{aligned} \tag{20}$$

The feasible set of (20) is still a polytope, and therefore we assume it can be distributively solved. Based on the obtained solution, we can generate a new cutting plane. The new cutting plane, together with previously obtained constraints, can help to attain a better approximation of the protection function. Notice, however, how to efficiently find a new cutting plane depends on which type of protection function we use. Details about how to find cutting planes for different convex set can be found in [14]. More importantly, for a collection of problems, we can attain a primal feasible solution as well as a dual upper bound to (2) via solving an approximating problem, as shown in the following theorem.

Theorem 1 *Assume that* $\bar{a}_{ji} \geq 0$, $\boldsymbol{x} \succeq \mathbf{0}$, *and* $g_j(\boldsymbol{x})$ *is a non-decreasing function in* x_i. *Let* $\bar{\boldsymbol{x}}^* \triangleq [\bar{x}_1^*, \bar{x}_2^*, ..., \bar{x}_N^*]^T$ *denote the optimal solution to the approximating problem in (20). For the* j^{th} *constraint, compute*

$$\delta_j = \min[1, \arg \max_{\delta \sum_i \bar{a}_{ji}\bar{x}_i^* + g_j(\delta \bar{\boldsymbol{x}}^*) \leq b_j, \delta \geq 0} (\delta)]. \tag{21}$$

In particular, for the ellipsoid, D-norm, and polyhedron uncertainty set,

$$\delta_j = \min \left[1, \frac{b_j}{\bar{\boldsymbol{a}}_j^T \bar{\boldsymbol{x}}^* + g_j(\bar{\boldsymbol{x}}^*)}\right]. \tag{22}$$

It is noticeable that δ_j *is an auxiliary variable associated with the* j^{th} *constraint in (3). Likewise, for the* i^{th} *variable, we can define an auxiliary variable* $\bar{\delta}_i$ *such that* $\bar{\delta}_i \triangleq \min_{\hat{a}_{ji}>0} \delta_j$. *Then* $\{\tilde{\boldsymbol{x}}^* \triangleq [\tilde{x}_1^*, \tilde{x}_2^*, ..., \tilde{x}_N^*]^T \ : \ \tilde{x}_i^* = \bar{x}_i^* \bar{\delta}_i\}$ *is a feasible solution to (3). Furthermore, the optimal objective function value of (3)* $f_0(\boldsymbol{x}^*)$ *is lower bounded by* $f_0(\tilde{\boldsymbol{x}}^*)$ *and upper bounded by* $f_0(\bar{\boldsymbol{x}}^*)$, *i.e.,*

$$f_0(\tilde{\boldsymbol{x}}^*) \leq f_0(\boldsymbol{x}^*) \leq f_0(\bar{\boldsymbol{x}}^*). \tag{23}$$

Proof Equation (22) follows directly from the fact that the protection function under ellipsoid, D-norm or polyhedron uncertainty set scales linearly with δ_j, i.e., $g_j(\delta_j \boldsymbol{x}) = \delta_j g_j(\boldsymbol{x})$. In addition, assume $\boldsymbol{x}$ is a feasible solution to (3), it follows from (19) that $\boldsymbol{x}$ is also a feasible solution to (20), as

$$\bar{\boldsymbol{a}}_j^T \boldsymbol{x} + \bar{g}_j(\boldsymbol{x}) \leq \bar{\boldsymbol{a}}_j^T \boldsymbol{x} + g_j(\boldsymbol{x}) \leq b_j, \ \forall j. \tag{24}$$

Hence, the feasible region of (3) is a subset of the feasible region of (20), which implies that $f_0(\boldsymbol{x}^*) \leq f_0(\bar{x}^*)$. Furthermore, for the j^{th} constraint of (3), we have

$$\begin{aligned} & \bar{\boldsymbol{a}}_j^T \tilde{\boldsymbol{x}}^* + g_j(\tilde{\boldsymbol{x}}^*) \\ = & \sum_i \bar{\delta}_i \bar{a}_{ji} \bar{x}_i^* + g_j(\tilde{\boldsymbol{x}}^*) && (25) \\ \leq & \sum_i \delta_j \bar{a}_{ji} \bar{x}_i^* + g_j(\delta_j \bar{\boldsymbol{x}}^*) && (26) \\ \leq & \ b_j. && (27) \end{aligned}$$

(25) holds as $\tilde{x}_i^* = \bar{x}_i^* \bar{\delta}_i$. (26) follows because $\forall \bar{a}_{ji} > 0$, $\delta_j \geq \bar{\delta}_i$, and $g_j(\tilde{\boldsymbol{x}}^*)$ is a non-decreasing function in $\tilde{x}_i^*$. (27) is obtained from the definition of δ_j. Based on (25) to (27) we conclude that $\tilde{\boldsymbol{x}}^*$ is a feasible solution to (3) and therefore $f_0(\boldsymbol{x}^*) \geq f_0(\tilde{\boldsymbol{x}}^*)$.

Comment 1 *To preserve the special structure of resulting problem we use a group of linear functions to approximate the protection function. However, it is clear from the proof that Theorem 1 is applicable as long as the condition in (19) is met. In other words, for any feasible approximating function* $\bar{g}_j(\boldsymbol{b}_j, \boldsymbol{x})$ *satisfying (19) (not necessarily linear), Theorem 1 implies that we can always compute an upper and lower bound to the optimal objective function of (3) by solving (20).*

Note that the key idea of the cutting-plane method is to choose a group of linear constraints to approximate the original problem in (3). We show next that for any given convex set, we can always find a group of linear constraints such that (19) is satisfied. Such a group of linear constraints is related to the well known *supporting hyperplanes*.

A hyperplane [17] is said to support a set S in Euclidean space $\mathbb{R}^n$ if it meets the following two conditions: 1) S is entirely contained in one of the two closed half-spaces determined by the hyperplane 2) S has at least one point on the hyperplane.

Supporting hyperplane theorem [17] is well known: If S is a closed convex set in Euclidean space $\mathbb{R}^n$, and $\boldsymbol{x}$ is a point on the boundary of S, then there exists a supporting hyperplane containing $\boldsymbol{x}$.

Since $g_j(\boldsymbol{x})$ is a convex function, the set $X_j \triangleq \{\boldsymbol{x} \succeq \boldsymbol{0} : \boldsymbol{a}_j^T \boldsymbol{x} + g_j(\boldsymbol{x}) \le b_j\}$ is a closed convex set. As a consequence, we obtain the following proposition on the existence of linear constraints satisfying (19).

Proposition 1 *Recall that* $X_j = \{\boldsymbol{x} \succeq \boldsymbol{0} : \boldsymbol{a}_j^T \boldsymbol{x} + g_j(\boldsymbol{x}) \le b_j\}$. *For any* $\boldsymbol{y} \succeq \boldsymbol{0}$ *and* $\boldsymbol{y} \notin X_j$. *There exists at least one* ***effective cutting plane*** $e_{j\ell}(\boldsymbol{x}) \triangleq \tilde{\boldsymbol{a}}_{j\ell}^T \boldsymbol{x}^* + \tilde{b}_{j\ell}$ *such that 1) (19) is satisfied, i.e.,* $e_{j\ell}(\boldsymbol{x}) \le g_j(\boldsymbol{x})$, $\forall \boldsymbol{x} \in X_j$ *; 2) the effective cutting plane contains at least one point in* X_j*; 3)* $e_{j\ell}(\boldsymbol{x})$ *separates* X_j *from* $\boldsymbol{y}$*, i.e.,* X_j *and* $\boldsymbol{y}$ *lie at different sides of* $e_{j\ell}(\boldsymbol{x})$ *and* $e_{j\ell}(\boldsymbol{x})$ *does not contain* $\boldsymbol{y}$.

Proof Since $\boldsymbol{y} \notin X_j$, and X_j is a closed convex set, we can project $\boldsymbol{y}$ onto the boundary of X_j in Euclidean space $\mathbb{R}^n$. Assume the projection is $\bar{\boldsymbol{y}}$. The supporting hyperplane theorem implies that we can find at least one supporting hyperplane containing $\bar{\boldsymbol{y}}$. In addition, X_j and $\boldsymbol{y}$ lie at different sides of this supporting hyperplane. Combing the fact that $\boldsymbol{y}$ does not belong to this hyperplane, we conclude that this supporting hyperplane is an effective cutting plane.

Based on the above theorem, we have the primal-dual cutting-plane method in the sequel, which stops when the gap between the upper-bound and lower-bound of the objection function is small enough.

Algorithm 1 *(Primal-dual cutting-plane method)*

1. *Set time k=0 and set the threshold* $\epsilon \ll 1$. *For each* j, *let* $\mathcal{H}_j$ *be the set containing the* j^{th} *constraint of the nominal problem.*
2. $k = k + 1$.
3. *Obtain an approximating problem (20) based on all the constraints in* $\{\mathcal{H}_j,\ 1 \le j \le M\}$.
4. *Solve the approximating problem. Calculate the sub-optimality gap* $\eta(k) = 1 - \frac{f_0(\tilde{\boldsymbol{x}}^*(k))}{f_0(\bar{\boldsymbol{x}}^*(k))}$. *If* $\eta(k) \le \epsilon$ *or* k *exceeds a given threshold, stop. Otherwise, for each* j, *if* $\bar{\boldsymbol{x}}(k)$ *violates the* j^{th} *constraint , then find an effective cutting plane, add it into the set* $\mathcal{H}_j$, *and go to step 2.*

Theorem 2 *The primal-dual cutting plane method monotonically converges to the optimal objective function value of (3). In addition, the sub-optimality gap converges to 0 when k goes to infinity, i.e.,* $\lim_{k\in\rightarrow\infty}\eta(k)=0$.

Proof Let $\mathcal{H}_j(k)$ and $X(k)$ denote respectively the value of $\mathcal{H}_j$ and the feasible region of approximating problem (20) at the k^{th} iteration of the above algorithm, we have $X(k+1)\subseteq X(k)$ as $\mathcal{H}_j(k)\subseteq\mathcal{H}_j(k+1)$. Since $f_0(\bar{\boldsymbol{x}}^*(k))=\max_{\boldsymbol{x}\in X(k)}f_0(\boldsymbol{x})$, we have

$$f_0(\bar{\boldsymbol{x}}^*(k))\geq f_0(\bar{\boldsymbol{x}}^*(k+1)),\forall k>1. \tag{28}$$

Hence the sequence $f_0(\bar{\boldsymbol{x}}^*(k))$ is monotonically non-increasing. In addition, since $X_j(k)\supseteq X_j$, $\forall k$, we have $\bar{X}_j\triangleq\lim_{k\rightarrow\infty}X_j(k)\supseteq X_j$. Assume the cutting-plane algorithm does not converge to the optimal solution to (3), in view of the fact that $X=\bigcap_{j=1}^{M}X_j$, there must exist one $\boldsymbol{y}\succeq\boldsymbol{0}$ in $\mathbb{R}^n$ and a $j\in\{1,...,n\}$ such that $\boldsymbol{y}\notin X_j$ but $\boldsymbol{y}\in\bar{X}_j$. On the other hand, it follows from Proposition 1 that any $\boldsymbol{y}\notin X_j$ can be removed from the set of feasible solution of (20) by adding an effective cutting plane, which contradicts the assumption and therefore conclude the proof.

Also, as $\bar{\boldsymbol{x}}^*(k)$ converges to $\boldsymbol{x}^*$, it follows from (21) that δ_j converges to 1, which implies that both $f_0(\tilde{\boldsymbol{x}}^*(k))$ and $f_0(\bar{\boldsymbol{x}}^*(k))$ converge to $f_0(\boldsymbol{x}^*)$, and therefore $\lim_{k\in\rightarrow\infty}\eta(k)=0$.

Comment 2 *The generation of the effective cutting plan is carried out on a row by row basis, and can be performed in either a parallel (synchronous) or serial (asynchronous) manner. As such, it provides a lot of flexibilities that leads to a distributed realization. We later give an example on applying the primal-dual cutting-plane method to the distributed rate control for service reliability in Section 3.*

Note that Proposition 1 not only proves the existence of an effective cutting plane but also gives an efficient way to calculate it. In practice, however, the operation of projection in Proposition 1 might pose difficulties for a distributed implementation, since it is a quadratic optimization problem. On the other hand, in many cases there exist more than one effective cutting planes for $\bar{\boldsymbol{x}}^*(k)$ [cf. Step 4 of Algorithm 1], we can then use an effective cutting plane different from the one proposed in Proposition 1 to simplify the computation. For instance, we will show how to simply generate an effective cutting plan for a D-norm protection function in Section 3 of this paper. Also, for the ellipsoid uncertainty set, [18] has proposed an efficient approach to find a good polytope approximation for a given ellipsoid. A drawback of the cutting-plane approach is that it may result in a large number of linear constraints and bring about a lot of extra message passings. In practice we can employ efficient numerical algorithms such as the active-set method to reduce the number of message passing.

Nonlinear Gauss-Seidel and Jacobi Method: The Gauss-Seidel (GS) and Jaobi methods were originally proposed as efficient means to solve linear systems, and have been extended to tackle unconstrained as well as constrained optimization problems [7]. GS methods and Jacobi methods have a variety of applications including CDMA multiuser detection [19] and decoding of low-density parity-check code [20]. The general principle of both methods is to decompose a multidimensional optimization problem into multiple one-dimensional optimization problems, and solve these one-dimensional optimization problems in a successive or parallel manner.

Let $x_i(k)$ denote the tentative value of the i^{th} variable obtained at the k^{th} iteration of the GS algorithm. Also let $X_i(\boldsymbol{x}_{-i})$ be the feasible region of the i^{th} variable when the values are other variables are set as $\boldsymbol{x}_{-i} = (x_1, ..., x_{i-1}, x_{i+1}, ..., x_N)$, i.e.,

$$X_i(\boldsymbol{x}_{-i}) \triangleq \left\{ x_i : \bar{\boldsymbol{a}}_j^T \boldsymbol{x} + g_j(\boldsymbol{x}) \ \leq b_j, \ \forall 1 \leq j \leq M \right\}, \tag{29}$$

where $\boldsymbol{x} = (x_i, \boldsymbol{x}_{-i})$. Then we can calculate $x_i(k+1)$ via the following GS iteration,

$$\begin{aligned} & x_i(k+1) \\ = \ & \arg \max_{x_i \in X_i(x_1(k+1), ..., x_{i-1}(k+1), x_{i+1}(k), ..., x_N(k))} f_0\left(x_1(k+1), ..., x_{i-1}(k+1), x_i, x_{k+1}(k), ..., x_N(k)\right). \end{aligned} \tag{30}$$

Similarily, in the Jacobi iteration, $x_i(k+1)$ is found by

$$\begin{aligned} & x_i(k+1) \\ = \ & \arg \max_{x_i \in X_i(x_1(k), ..., x_{i-1}(k), x_{i+1}(k), ..., x_N(k))} f_0\left(x_1(k), ..., x_{i-1}(k), x_i, x_{i+1}(k), ...x_N(k)\right). \end{aligned} \tag{31}$$

The difference between the GS method and the Jacobi method lies in the order of solving the one-dimensional problems. The GS method solves the one-dimensional problems sequentially, while the Jacobi method solve in parallel. They can also be combined together to form hybrid algorithms. We later show a distributed robust power control application, which is closely related to the nonlinear GS and Jacobi methods.

3 Robust Multipath Rate Control for Service Reliability

3.1 Nominal and Robust Formulations

Consider a wireline network where some links might fail because of human mistakes, software bugs, hardware defects, or natural hazard. Network operators typically reserve some bandwidth for some backup paths. When the primary paths fail, some or all of the traffic will be re-routed to the corresponding disjoint backup paths. Fast system recovery schemes are essential to ensure service availability in the presence of link failure. There are three key components for fast system recovery [21]: identifying a backup path disjoint from the primary path, computing network resource (such as bandwidth) in reservation prior to link failure, and detecting the link failure in real-time and re-route the traffic. The first component has been investigated extensively in graph theory. The third component has been extensively studied in system research community. Here we consider the robust rate control and bandwidth reservation in the face of possible failure of primary path, which is related to the second component.

We first consider the nominal problem with no link failures. Following similar notation as in Kelly's seminal work [11], we consider a network with S users, L links and T paths, indexed by s, l and t, respectively. Each user is a unique flow from one source node to one destination node. There could be multiple users between the same source-destination node pair. The network is characterized by the $L \times T$ path-availability $0-1$ matrix

$$[\boldsymbol{D}]_{lt} = \begin{cases} d_{lt} = 1, & \text{if link } l \text{ is on path } t, \\ 0, & \text{otherwise.} \end{cases}$$

and $T \times S$ primary-path-choice nonnegative matrix

$$[\boldsymbol{W}]_{ts} = \begin{cases} w_{ts}, & \text{if user } s \text{ uses path } t \text{ as the primary path,} \\ 0, & \text{otherwise.} \end{cases}$$

where $w_{ts} > 0$ indicates the percentage that user s allocates its rate to primary path t, and $\sum_t w_{ts} = 1$. Let $\boldsymbol{x}$, $\boldsymbol{c}$, and $\boldsymbol{y}$ denote source rates, link capacities, and aggregated path rates, respectively. The nominal multi-path rate control problem is

$$\begin{aligned} \text{maximize} \quad & \sum_s f_s(x_s) \\ \text{subject to} \quad & \boldsymbol{D}\boldsymbol{y} \preceq \boldsymbol{c}, \quad \boldsymbol{W}\boldsymbol{x} \preceq \boldsymbol{y}, \\ \text{variables} \quad & \boldsymbol{x} \succeq \boldsymbol{0}, \boldsymbol{y} \succeq \boldsymbol{0}, \end{aligned} \tag{32}$$

where $f_s(x_s)$ is user s' utility function, which is increasing and strictly concave in x_s.

To represent Problem (32) in a more compact way, we denote $\boldsymbol{R} = \boldsymbol{D}\boldsymbol{W}$ as the link-source matrix

$$[\boldsymbol{R}]_{ls} = \begin{cases} r_{ls} = \sum_{t \in T(l)} w_{ts}, & \text{if user } s \text{ uses link } l \text{ in one of its primary paths,} \\ 0, & \text{otherwise.} \end{cases}$$

where $T(l)$ denotes the set of all paths associated with link l, i.e., $T(l) = \{t : d_{lt} = 1\}$.

The nominal problem can be compactly rewritten as

$$\begin{aligned} \text{maximize} \quad & \sum_s f_s(x_s) \\ \text{subject to} \quad & \boldsymbol{R}\boldsymbol{x} \preceq \boldsymbol{c}, \\ \text{variables} \quad & \boldsymbol{x} \succeq \boldsymbol{0}\ . \end{aligned} \tag{33}$$

To ensure robust data transmission against the link failures, each user also determines a backup path when it joins the network. The nonnegative backup path choice matrix is

$$[\boldsymbol{B}]_{ts} = \begin{cases} b_{ts}, & \text{if user } s \text{ uses path } t \text{ as the backup path,} \\ 0, & \text{otherwise.} \end{cases}$$

where $b_{ts} > 0$ indicates the maximum percentage that user s allocates its rate to path t. The actual rate allocation will be a random variable between 0 and b_{ts}, depending on whether the primary paths fail. We further assume that a path can only be used as either a primary path or a backup path for the same user but not both. The corresponding *robust multi-path routing rate allocation problem* is given by

$$\begin{aligned} \text{maximize} \quad & \sum_s f_s(x_s) \\ \text{subject to} \quad & \sum_{s \in S(l)} r_{ls} x_s + \sum_{t \in T(l)} g_t(\boldsymbol{b}_t, \boldsymbol{x}) \leq c_l, \quad \forall l. \\ \text{variables} \quad & \boldsymbol{x} \succeq \boldsymbol{0}\ . \end{aligned} \tag{34}$$

Here $S(l)$ denotes the set of users using the link l in one of their primary paths, and $\sum_{s \in S(l)} r_{ls} x_s$ is the aggregate rate from users. Moverover, $g_t(\boldsymbol{b}_t, \boldsymbol{x})$ is the protection function for the traffic from users who use path t as their backup path, and $\boldsymbol{b}_t$ is the t^{th} row of matrix $\boldsymbol{B}$.

There are many ways of characterizing the protection function. Here we take D-norm as an example. Let $\mathcal{E}_t = \{s : b_{ts} > 0, \forall s\}$ denote the set of users who utilize path t as the backup path, and $\mathcal{F}_{t,\Gamma_t}$ denote a set such that

$$\mathcal{F}_{t,\Gamma_t} \subseteq \mathcal{E}_t \text{ and } |\mathcal{F}_{t,\Gamma_t}| = \Gamma_t.$$

Here Γ_t denotes the number of users who might experience path failures, and its value controls the tradeoff between robustness and performance. The protection function can then be written as

$$g_t(\boldsymbol{b}_t, \boldsymbol{x}) = \max_{\mathcal{F}_{t,\Gamma_t} \subseteq \mathcal{E}_t} \sum_{s \in \mathcal{F}_{t,\Gamma_t}} b_{ts} x_s, \forall t. \tag{35}$$

Notice that Γ_t is a parameter (instead of a variable) in the protection function (35).

A centralized algorithm such as the interior point method can be used to solve the service reliability problem (34). In practice, however, a distributed solution is preferred in many situations. For example, for a large-scale communication network, multiple nodes and associated links might be removed or updated, and it could be difficult to collect the updated topology information of the whole network. As such, a centralized algorithm may not be applicable. Furthermore, a distributed algorithm can be used to dynamically adjust the rates according to changes in the network topology. Such scenarios motivate us to seek a distributed solution of the multipath rate control problem. Since we consider service reliability in this application, most likely such a update is only required infrequently.

3.2 Distributed Primal-dual Algorithms

We now develop a fast distributed algorithm to to solve the robust optimization of multipath rate control based on a combination of active-set method [22] and dual-based decomposition method. The proposed algorithm can be carried out in a fully distributed manner.

We first show that the nonlinear constraints in Problem (34) can be replaced by a set of linear constraints:

Proposition 2 *For any path t, the single constraint*

$$\sum_{s \in S(l)} r_{ls} x_s + \sum_{t \in T(l)} g_t(\boldsymbol{b}_t, \boldsymbol{x}) \leq c_l, \tag{36}$$

is equivalent to the following set of constraints

$$\sum_{s \in S(l)} r_{ls} x_s + \sum_{t \in T(l)} \sum_{s \in \mathcal{F}_{t,\Gamma_t}} b_{ts} x_s \leq c_l, \forall \left(\mathcal{F}_{t,\Gamma_t},\ \forall t \in T(l)\right) \text{ such that } \mathcal{F}_{t,\Gamma_t} \subseteq \mathcal{E}_t \tag{37}$$

Proof Let $S(l)$ denote the set of users using link l on their primary paths. From (35), we know $\sum_{s \in \mathcal{F}_{t,\Gamma_t}} b_{ts} x_s \leq g_t(\boldsymbol{b}_t, \boldsymbol{x})$ for all $\mathcal{F}_{t,\Gamma_t} \subseteq \mathcal{E}_t$. If $\boldsymbol{x}^*$ satisfies constraint (36), we have

$$\sum_{s \in S(l)} r_{ls} x_s^* + \sum_{t \in T(l)} \sum_{s \in \mathcal{F}_{t,\Gamma_t}} b_{ts} x_s^* \leq \sum_{s \in S(l)} r_{ls} x_s^* + \sum_{t \in T(l)} g_t(\boldsymbol{b}_t, \boldsymbol{x}^*) \leq c_l, \tag{38}$$

i.e., it also satisfies the set of constraints in (37). On the other hand, if (37) is true, then

$$\begin{aligned}
&\sum_{s\in S(l)} r_{ls}x_s + \sum_{t\in T(l)} g_t(\boldsymbol{b}_t, \boldsymbol{x}) \\
&= \sum_{s\in S(l)} r_{ls}x_s + \sum_{t\in T(l)} \max_{\mathcal{F}_{t,\Gamma_t}\subseteq\mathcal{E}_t} \sum_{s\in\mathcal{F}_{t,\Gamma_t}} b_{ts}x_s \qquad (39)\\
&\le \max_{(\mathcal{F}_{t,\Gamma_t},\, \forall t\in T(l)):\mathcal{F}_{t,\Gamma_t}\subseteq\mathcal{E}_t} \left(\sum_{s\in S(l)} r_{ls}x_s^* + \sum_{t\in T(l)} \sum_{s\in\mathcal{F}_{t,\Gamma_t}} b_{ts}x_s \right) \le c_l. \qquad (40)
\end{aligned}$$

Therefore these two constraints are equivalent.

We further define some short-hand notations. For each choice of paramters ($|T(l)|$ sets of users)

$$\boldsymbol{\mathcal{F}}_l = \left(\mathcal{F}_{t,\Gamma_t} \subseteq \mathcal{E}_t,\ \forall t \in T(l)\right),$$

we define a set corresponding to the choices of paths and users:

$$\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l} \triangleq \left\{(t,s) : t \in T(l), s \in \mathcal{F}_{t,\Gamma_t}\right\}.$$

We further define $\mathcal{G}_l = \{\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}, \forall\boldsymbol{\mathcal{F}}_l\}$. The constraints in (37) can be rewritten as follows,

$$\sum_{s\in S(l)} r_{ls}x_s + \sum_{(t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}} b_{ts}x_s \le c_l,\ \forall\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l} \in \mathcal{G}_l. \qquad (41)$$

Note the number of constraints in (41) is $\prod_{t\in T(l)} \binom{|\mathcal{E}_t|}{\Gamma_t}$, and increases quickly with Γ_t and $|\mathcal{E}_t|$. This motives us to design an alternative method to solve (34). We next present a sequential optimization approach. The basic idea is to iteratively generate a set $\bar{\mathcal{G}}_l \subseteq \mathcal{G}_l$, and use the following set of constraints to approximate (41):

$$\sum_{s\in S(l)} r_{ls}x_s + \sum_{(t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}} b_{ts}x_s \le c_l,\ \forall\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l} \in \bar{\mathcal{G}}_l. \qquad (42)$$

This leads to a relaxation of Problem (34):

$$\begin{aligned}
&\text{maximize} \quad \sum_s f_s(x_s) \qquad (43)\\
&\text{subject to} \quad \sum_{s\in S(l)} r_{ls}x_s + \sum_{(t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}} b_{ts}x_s \le c_l,\ \forall\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l} \in \bar{\mathcal{G}}_l,\quad \forall l,\\
&\text{variables} \quad \boldsymbol{x} \succeq \mathbf{0}\,.
\end{aligned}$$

Let $\bar{\boldsymbol{x}}$ denote an optimal solution to the relaxed problem (43) and $\boldsymbol{x}^*$ denote an optimal solution of the original problem (34). If $\bar{\mathcal{G}}_l = \mathcal{G}_l$, then we have $\sum_s f_s(\bar{\boldsymbol{x}}_s) = \sum_s f_s(\boldsymbol{x}_s^*)$. Even if $\bar{\mathcal{G}}_l \subset \mathcal{G}_l$, it is still possible that the two optimal objective values are the same as shown in the following theorem:

Theorem 3 $\sum_s f_s(\bar{x}_s) = \sum_s f_s(x_s^*)$ *if the following condition holds*

$$\max_{\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}\in\bar{\mathcal{G}}_l} \sum_{(t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}} b_{ts}\bar{x}_s = \max_{\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}\in\mathcal{G}_l} \sum_{(t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}} b_{ts}\bar{x}_s,\ \forall l. \qquad (44)$$

Proof Since $\bar{\mathcal{G}}_l \subseteq \mathcal{G}_l$ and $\bar{\boldsymbol{x}}$ is the optimal source rate allocation of the relaxed problem in (43), we have $\sum_s f_s(\bar{x}_s) \geq \sum_s f_s(x_s^*)$. However, the condition in (44) implies that

$$\begin{aligned} &\sum_s r_{ls}\bar{x}_s + \max_{\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l} \in \bar{\mathcal{G}}_l} \sum_{(t,s)\in\mathcal{F}_l} b_{ts}\bar{x}_s \\ =&\sum_s r_{ls}\bar{x}_s + \max_{\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l} \in \mathcal{G}_l} \sum_{(t,s)\in\mathcal{F}_l} b_{ts}\bar{x}_s \leq c_l, \ \forall l, \end{aligned} \tag{45}$$

hence $\bar{\boldsymbol{x}}$ is a feasible solution of (34), and $\sum_s f_s(\bar{x}_s) \leq \sum_s f_s(x_s^*)$. Thus we have $\sum_s f_s(\bar{x}_s) = \sum_s f_s(x_s^*)$.

Next we develop a distributed algorithm (Algorithm 2) to solve Problem (43) for a fixed $\bar{\mathcal{G}}_l$ for each l, which is suboptimal for solving Problem (34). We then design an optimal distributed algorithm (Algorithm 3) that achieves the optimal solution of Problem (34) by iteratively using Algorithm 2.

By relaxing the constraints in Problem (43) using dual variables $\boldsymbol{\lambda} = \{\{\lambda_{li}\}_{i=1}^{|\bar{\mathcal{G}}_l|}\}_{l=1}^{L}$, we obtain the following Lagrangian,

$$\bar{Z}(\boldsymbol{\lambda}, \boldsymbol{x}) = \sum_s f_s(x_s) + \sum_l \sum_{i=1}^{|\bar{\mathcal{G}}_l|} \lambda_{li} \left(c_l - \sum_{s\in S(l)} r_{ls}x_s - \sum_{(t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}(i)} b_{ts}x_s \right).$$

For easy indexing, here we denote each set $\bar{\mathcal{G}}_l = \left\{\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}(i), \ i = 1, \ldots, |\bar{\mathcal{G}}_l|\right\}$.

The dual function is

$$Z(\boldsymbol{\lambda}) = \max_{\boldsymbol{x}\succeq\boldsymbol{0}} \bar{Z}(\boldsymbol{\lambda}, \boldsymbol{x}). \tag{46}$$

The optimization over $\boldsymbol{x}$ in (46) can be decomposed into one problem for each user s:

$$\max_{x_s\geq 0} \left(f_s(x_s) - \sum_l \sum_{i=1}^{|\bar{\mathcal{G}}_l|} \left(\lambda_{li} r_{ls} + \lambda_{li} \sum_{(t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}(i)} b_{ts} \right) x_s \right). \tag{47}$$

Notice that link l can be viewed as the composition of $|\bar{\mathcal{G}}_l|$ sub-links and each sub-link is associated with a price (dual variable) λ_{li}. Each user s determines its transmission rate x_s by considering prices from both its primary path and backup path.

The master dual problem is

$$\min_{\boldsymbol{\lambda}\succeq\boldsymbol{0}} Z(\boldsymbol{\lambda}), \tag{48}$$

which can be solved by the subgradient method. For each dual variable λ_{li}, its subgradient can be calculated as

$$\zeta_{li}(\lambda_{li}) = c_l - \sum_{s\in S(l)} r_{ls}x_s - \sum_{(t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}(i)} b_{ts}x_s. \tag{49}$$

The value of λ_{li} will be updated using the subgradient information correspondingly. The complete algorithm is given as in Algorithm 2.

Algorithm 2 *(Distributed primal-dual subgradient algorithm)*

1. *Set time* $k = 0$, $\boldsymbol{\lambda}(0) = \boldsymbol{0}$, $\boldsymbol{\mu}(0) = \boldsymbol{0}$, *and thresholds* $\boldsymbol{\epsilon} \ll \boldsymbol{1}$.

2. *Let* $k = k + 1$.
3. *Each user* s *determines* $x_s(k)$ *by solving Problem (47).*
4. *Each user* s *passes the value of* $x_s(k)$ *to each link associated with this user.*
5. *Each link* l *calculates the subgradients* $\boldsymbol{\zeta}_l(\boldsymbol{\lambda}_l(k)) = \{\zeta_{li}(\lambda_{li}(k)), \forall t, i\}$ *as in (49). and* $\bar{\Delta}_s(l,k) = \frac{c_l}{\bar{c}_l}$, *where* $\bar{c}_l = \max_{\mathcal{F}_l \in \bar{\mathcal{G}}_l}\left(\sum_{s\in S(l)} r_{ls}x_s(k) + \sum_{(t,s)\in\mathcal{F}_l} b_{ts}x_s(k)\right)$.
6. *Each link* l *updates the dual variables* $\boldsymbol{\lambda}_l(k+1) = \max\{\boldsymbol{\lambda}_l(k) + \theta(k)\boldsymbol{\zeta}_l(\boldsymbol{\lambda}_l(k)), 0\}$.
7. *Each user* s *who uses link* l *on either its primary or backup paths calculates the associated dual prices by passing messages over path* t *from its destination to its source. User* s *also calculates* $\bar{\Delta}_s(k) = \min_l\left[\left(\bar{\Delta}_s(l,k)\right)_{r_{ls}>0}, \ \left(\bar{\Delta}_s(l,k)\right)_{b_{ts}>0,\ (t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}(i)}\right]$ *and* $\bar{x}_s(k) = x_s(k)\bar{\Delta}_s(k)$*by collecting messages from its associated links. If* $\eta_i(k) = 1 - \frac{\sum_s f_s(\bar{x}_s(k))}{Z(\boldsymbol{\lambda}_k)} \le \epsilon_i$ *(where* $Z(\boldsymbol{\lambda}_k)$ *is given in (46)), stop. Otherwise go to step 2.*

Here $\theta(k)$ is the step-size at time k. If it meets conditions such as those given in [16, pp. 505], then this subgradient method is guaranteed to converge to the optimal solution. Common choices of stepsize include the constant step size ($\theta(k) = \beta$), diminish stepsize ($\theta(k) = \frac{\beta}{\sqrt{k}}$), and square summable ($\theta(k) = \frac{\beta}{k}$). The parameter ϵ_i is a pre-defined small threshold. In addition, due to the special structure of (43), we can characterize the sub-optimality gap of the obtained solution at each iteration, as shown below.

Theorem 4 *Let* $P^*(\{\bar{\mathcal{G}}_l\})$ *be the optimal object function value of the relaxed problem (43),* $\boldsymbol{\lambda}^k$ *be the dual variables at the* k^{th} *iteration of the subgradient algorithm 2, and* $[\bar{x}_1(k), ..., \bar{x}_S(k)]$ *be a feasible solution to (43). Then* $P^*(\{\bar{\mathcal{G}}_l\})$ *is lower bounded by* $\sum_s f_s(\bar{x}_s(k))$ *and upper bounded by* $Z(\boldsymbol{\lambda}^k)$, *i.e.,*

$$\sum_s f_s(\bar{x}_s(k)) \le P^*(\{\bar{\mathcal{G}}_l\}) \le Z(\boldsymbol{\lambda}^k). \tag{50}$$

Furthermore, the sub-optimality gap $\eta_i(k)$ *satisfies* $\lim_{k\to\infty}\eta_i(k) = 0$ *for all* i.

Proof For the l^{th} link, we have

$$\max_{\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}\in\bar{\mathcal{G}}_l}\left(\sum_{s\in S(l)} r_{ls}\bar{x}_s(k) + \sum_{(t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}} b_{ts}\bar{x}_s(k)\right)$$

$$= \max_{\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}\in\bar{\mathcal{G}}_l}\left(\sum_{s\in S(l)} r_{ls}x_s(k)\bar{\Delta}_s(k)x_s(k) + \sum_{(t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}} b_{ts}x_s(k)\bar{\Delta}_s(k)\right) \tag{51}$$

$$\le \max_{\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}\in\bar{\mathcal{G}}_l}\left(\sum_{s\in S(l)} r_{ls}x_s(k)\Delta_s(l,k)x_s(k) + \sum_{(t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}} b_{ts}x_s(k)\Delta_s(l,k)\right) \tag{52}$$

$$= \Delta_s(l,k)\max_{\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}\in\bar{\mathcal{G}}_l}\left(\sum_{s\in S(l)} r_{ls}x_s(k)x_s(k) + \sum_{(t,s)\in\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}} b_{ts}x_s(k)\right) \tag{53}$$

$$= \frac{c_l}{\bar{c}_l} \times \bar{c}_l = c_l. \tag{54}$$

(51) follows from the definition of $\bar{\boldsymbol{x}}_s(k)$. (52) holds as $\bar{\Delta}_s(k) \le \bar{\Delta}_s(l,k)$, $\forall r_{ls} > 0, \forall b_{ts} > 0,\ (t,s) \in \mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}(i)$. Hence, $[\bar{x}_1(k), ..., \bar{x}_S(k)]$ is a feasible solution to (43) and

$\sum_s f_s(\bar{x}_s(k)) \leq P^*(\{\bar{\mathcal{G}}_l\})$. The second part of the inequality (50) follows directly from the weak duality theorem, i.e.,

$$Z(\boldsymbol{\lambda}^k) \geq \min_{\boldsymbol{\lambda}} Z(\boldsymbol{\lambda}) \geq P^*(\{\bar{\mathcal{G}}_l\}). \tag{55}$$

If a proper step-size is chosen, e.g., constant or diminish step size, the subgradient method converges to the optimal solution [16], and therefore $\lim_{k\to\infty} \sum_s f_s(\bar{x}_s(k)) = P^*(\bar{\mathcal{G}}_l)$. This combining with the strong duality theorem (i.e., $P^*(\bar{\mathcal{G}}_l) = \min_{\boldsymbol{\lambda}} Z(\lambda)$) implies that $\lim_{k\to\infty} 1 - \frac{\sum_s f_s(\bar{x}_s(k))}{Z(\boldsymbol{\lambda}_k)} = 0$.

Comment 3 *The primary goal of computing $\eta_i(k)$ is to estimate the gap between the current solution and the optimal one. In order to reduce the amount of message-passing, we can calculate $\eta_i(k)$ less frequently, e.g., every 20 iterations. Also, $\bar{\Delta}_s(k)$ approaches 1 when the sub-optimality gap converges to zero. Hence each user s could use it as a metric to measure the near-optimality of the current feasible solution, which can be implemented in a distributed manner.*

During each iteration in Algorithm 2, we can calculate a lower bound to the optimal objective function value of the original problem (34) as stated in Theorem 5. The proof is similar to that of Theorem 4 and is thus omitted.

Theorem 5 *At the k^{th} step of Algorithm 2, let $\Delta_s(k) \triangleq \min_{l, r_{ls}>0} \frac{c_l}{\hat{c}_l(k)}$, where*

$$\hat{c}_l(k) = \max_{\mathcal{Q}_{\mathcal{F}_l} \in \mathcal{G}_l} \left(\sum_s r_{ls} x_s(k) + \sum_{(t,s)\in \mathcal{Q}_{\mathcal{F}_l}} b_{ts} x_s(k) \right). \tag{56}$$

The optimal objective function value of (34) is upper bounded by $Z(\boldsymbol{\lambda}^k)$ and lowered bounded by $\sum_s f_s(\tilde{x}_s(k))$, where $\tilde{x}_s(k) = x_s(k)\Delta_s(k)$ for each user s.

Furthermore, the sub-optimality gap of the solution $[\tilde{x}_1(k), ..., \tilde{x}_S(k)]$ can be defined as $\eta_o(k) = 1 - \frac{\sum_s f_s(\tilde{x}_s(k))}{Z(\boldsymbol{\lambda}^k)}$. Based on Algorithm 2, we propose the optimal distributed algorithm 3 to find the optimal solution of Problem (34). The basic idea of Algorithm 3 is to dynamically generate a set of active constraints to approximate the D-norm protection function. This method proceeds in a iterative manner.At each iteration, we use a subset of it to approximate the whole constraint set $\mathcal{G}_l = \{\mathcal{Q}_{\mathcal{F}_l}\}$, and Algorithm 2 is invoked to solve the relaxed problem.

Algorithm 3 *(Distributed primal-dual cutting-plane algorithm for robust multipath rate control)*

1. *For each link l, let $\bar{\mathcal{G}}_l = \{\}$.*
2. *Set the time $k_o = 0$, and the sub-optimality gap for the outer loop iteration $\epsilon_o \ll 1$.*
3. *Let $k_o = k_o + 1$.*
4. *Set the threshold of sub-optimality gap for the inner loop iteration $\epsilon_i(k_o) \ll 1$.*
5. *Run Algorithm 2 until $\eta_i(k_i) \leq \epsilon_i(k_o)$ or the number of iterations k_i exceeds a pre-specified threshold. If the sub-optimality gap $\eta_o(k_i) \leq \epsilon_o$ or k_o exceeds a pre-specified threshold, stop. Otherwise, go to the step 6.*
6. *Each user s passes the tentative data rate x_s^* to every link associated with this user.*
7. *For each path t, rank $\{b_{ts}x_s^*\}_{s\in\mathcal{E}_t}$ in descending order and take the Γ_t largest items to construct a new set $\mathcal{F}_{t,\Gamma_t}$.*

8. *For each link l, construct set $\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l} = \{(t,s) : t \in T(l), s \in \mathcal{F}_{t,\Gamma_t}\}$ and add it into $\bar{\mathcal{G}}_l$. Go to step 3.*

Algorithm 3 iteratively generates a group of relaxed problems to approximate the original problem (34), and eventually converges to the optimal solution if both ϵ_i and ϵ_o are set as 0. Furthermore, the tradeoff between the performance and complexity can be controlled via setting ϵ_i and ϵ_o at different values. An intuitive way to accelerate the convergence speed of Algorithm 3 is to set ϵ_i as a relatively large value when k_o is small and then gradually decreases with k_o. As such, the algorithm can quickly find out useful cutting planes at the first several iterations and then gradually converge to the optimal solution. Notice that the amount of message passing at each step of Algorithm 2 increases with $|\bar{\mathcal{G}}_l|$. On the other hand, due to the special structure of the protection function (35), a lot of constraints in $\bar{\mathcal{G}}_l$ should be inactive. This motivates us to consider the following distributed active-set method. We first show that inactive constraints are redundant in the theorem below.

Theorem 6 *Assuming the optimal solution to the relaxation problem (43) is $\boldsymbol{x}^* \triangleq [x_1^*, x_2^*, ...x_S^*]^T$. For the set $\mathcal{G}_l$ of each link l, if we remove the subset $\{\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}\}$ corresponding to the inactive constraints at the optimal solution $\boldsymbol{x}^*$, i.e.,*

$$\left\{ \mathcal{Q}_{\boldsymbol{\mathcal{F}}_l} : \mathcal{Q}_{\boldsymbol{\mathcal{F}}_l} \in \bar{\mathcal{G}}_l, \sum_{s \in S(l)} r_{ls} x_s^* + \sum_{(t,s) \in \mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}} b_{ts} x_s^* < c_l \right\}, \quad \forall l, \tag{57}$$

from problem (43), the resulting problem has the same optimal solution as (43).

Proof Assume that at least one constraint in Problem (43) is not active at its optimal solution $\boldsymbol{x}^*$. Without loss of generality, assume that such inactive constraint corresponds to set $\mathcal{Q}_{\boldsymbol{\mathcal{F}}_{\bar{l}}}(j) \in \bar{\mathcal{G}}_{\bar{l}}$. Then we can remove the corresponding constraint and looks at the following problem:

$$\begin{aligned}
&\text{maximize} && \sum_s f_s(x_s) && (58)\\
&\text{subject to} && \sum_{s \in S(l)} r_{ls} x_s + \sum_{(t,s) \in \mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}(i)} b_{ts} x_s \le c_l, \ 1 \le i \le |\bar{\mathcal{G}}_l|, \ l \ne \bar{l}\,, \\
&&& \sum_{s \in S(\bar{l})} r_{\bar{l}s} x_s + \sum_{(t,s) \in \mathcal{Q}_{\boldsymbol{\mathcal{F}}_{\bar{l}}}(i)} b_{ts} x_s \le c_{\bar{l}}, \ \ 1 \le i \le |\bar{\mathcal{G}}_{\bar{l}}|, \ \ i \ne j, \\
&\text{variables} && \boldsymbol{x} \succeq \mathbf{0}\,.
\end{aligned}$$

It remains to show that $\boldsymbol{x}^*$ is the optimal solution of (58). Notice that the objective function of (43) is continuously differentiable and concave, and the constraints in (43) are continuously differentiable and convex functions, the following KKT conditions are sufficient and necessary for optimality,

$$f_s'(x_s^*) - \sum_l \sum_{i=1}^{|\bar{\mathcal{G}}_l|} \lambda_{li}^* \left(r_{ls} + \sum_{(t,s)\in \mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}(i)} b_{ts} \right) = 0, \tag{59}$$

$$\sum_s r_{ls} x_s^* + \sum_{(t,s)\in \mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}(i)} b_{ts} x_s^* \le c_l, \tag{60}$$

$$\lambda_{li}^*(c_{li} - r_{ls}x_s^* - \sum_{(t,s)\in \mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}(i)} b_{ts}x_s^*) = 0,\ 1 \le i \le |\bar{\mathcal{G}}_l|,\ \forall l, \tag{61}$$

$$\lambda_{li}^* \ge 0,\ 1 \le i \le |\bar{\mathcal{G}}_l|,\ \forall l, \tag{62}$$

where $f_s'(x)$ denotes the first order derivative of the function $f_s(x)$. Since

$$\sum_s r_{\bar{l}s} x_s^* + \sum_{(t,s)\in \mathcal{Q}_{\boldsymbol{\mathcal{F}}_{\bar{l}}}(j)} b_{ts} x_s^* < c_l, \tag{63}$$

it follows from (61) that $\lambda_{\bar{l}j}^* = 0$. This implies that for the solution $\boldsymbol{x}^*$, we can find a group of dual variables $\{\lambda_{li}^* \ge 0,\ li \ne \bar{l}j\}$ to satisfy the KKT condition of (58). Consequently, $\boldsymbol{x}^*$ is the optimal solution to (58).

Using induction, we can show that the optimal solution remains the same if we remove all inactive constraints.

Theorem 6 motivates us to design a new algorithm that reduces the amount of message passing in the distributed algorithm, as shown below.

Algorithm 4 *(Distributed primal-dual active-set algorithm for robust multipath rate control)*

1. *Each link l sets $\bar{\mathcal{G}}_l$ as an empty set.*
2. *Set the number of outer loop iteration $k_o = 0$, and the threshold of sub-optimality gap for the outer loop iteration $\epsilon_o \ll 1$.*
3. *Let $k_o = k_o + 1$.*
4. *Set the sub-optimality gap for the inner loop iteration $\epsilon_i(k_o) \ll 1$.*
5. *Algorithm 2 is carried out until $\eta_i(k_i) \le \epsilon_i(k_o)$ or the number of iterations k_i exceeds a pre-specified threshold. If the sub-optimality gap $\eta_o(k_i) \le \epsilon_o$ or k_o exceeds a pre-specified threshold, stop. Otherwise, go to the step 6.*
6. *Each user s passes the tentative data rate x_s^* to every link associated with this user.*
7. *Rank $\{b_{ts}x_s^*\}_{s\in\mathcal{E}_t}$ in descending order for each path t, and take the Γ_t largest items to construct a new set $\mathcal{F}_{t,\Gamma_t}$.*
8. *For each link l, construct set $\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l} = \{(t,s) : t \in T(l), s \in \mathcal{F}_{t,\Gamma_t}\}$. Remove the redundant constraint set from $\bar{\mathcal{G}}_l$ according to (57), and add $\mathcal{Q}_{\boldsymbol{\mathcal{F}}_l}$ into $\bar{\mathcal{G}}_l$. Go to step 3.*

We next prove that the above algorithm converges to the optimal solution of Problem (34).

Theorem 7 *Assume the objective function $\sum_s f_s(x_s)$ is strictly concave. The distributed active-set algorithm 4 converges globally and monotonically to the optimal solution of the robust multipath rate control problem in (34) in finite steps.*

Proof Let $\boldsymbol{x}^*(q) \triangleq [x_1^*(q), x_2^*(q), x_3^*(q), ..., x_S^*(q)]^T$ be the solution obtained at the q^{th} step of the active-set method. We first prove that if the active-set method does not converge to the optimal solution to (34) at the q^{th} step, we have $\sum_s f_s(x^*(q)) > \sum_s f_s(x^*(q+1))$. It follows from Theorem 6 that $\boldsymbol{x}^*(q)$ is the optimal solution to the relaxed problem in (43) with only active constraints. Since the objective function is strictly concave, if $\boldsymbol{x}^*(q)$ is not an optimal solution to (34), it must violate one of the constraints in (37). Consequently we can add a cutting plane into the constraint set and remove $\boldsymbol{x}^*(q)$ from the feasible set of the relaxed problem at the $(q+1)^{th}$ step of the active-set method. This combing with the fact that the objective function is strictly concave implies $\sum_s f_s(x^*(q)) > \sum_s f_s(x^*(q+1))$, i.e., the objective function value of the active-set method is monotonically decreasing before its convergence.

Let $\mathcal{C}(q)$ be the set of active constraints at the q^{th} step of the distributed primal-dual active-set method. Assume the distributed active-set method does not converge before the q^{th} step, we have $\sum_s f_s(x^*(1)) > \sum_s f_s(x^*(2)) > \cdots > \sum_s f_s(x^*(q))$. Since the objective function is strictly concave,we have $\mathcal{C}(\bar{q}) \neq \mathcal{C}(q),\ \forall \bar{q} < q$, which implies that

$$\mathcal{C}(1) \neq \mathcal{C}(2) \neq \cdots \neq \mathcal{C}(q). \tag{64}$$

In view of the fact that the number of different $\mathcal{C}(q)$ is finite [cf. (37)], we conclude that the proposed algorithm is guaranteed to converge when q goes to infinity.

3.3 Numerical Results

Here we consider a simple network model with three nodes, 13 links and 13 paths, as shown in Fig. 1. Paths $1-12$ are single link paths that use links $1-12$, respectively. Path 13 consists of links 12 and 13. The first 11 paths are used as primary paths by 11 users in the network. Path 12 is used as the backup path by users $1-8$, and path 13 is used as the backup path by users $9-11$. Each user s has a logarithmic utility function $\log(x_s)$, where the unit of x_s is bps. The capacity of each link is fixed at 1 Mbps.

Figure 2 shows the convergence behavior of the Algorithm 4. Here $\Gamma_{12} = \Gamma_{13} = 3$, which means we allow maximum 3 and 3 users to experience failures and use path 12 and path 13 as their backup paths to transmit their data respectively. Two variations of active-set algorithms are considered in Figure 2. We call the first type of active-set method fixed active-set method, as we assume fixed number of *inner iterations*, namely, the iterations of the subgradient method, at each stage. The algorithm stops when $\eta_o \leq \epsilon_o$. The number of inner iterations is set as 100, 50, 15, respectively, corresponding to the first three subplots in Figure 2. In the second version of active-set method, i.e., the adaptive active-set method, the estimated sub-optimality gap, i.e., η_i is are used as the stopping criteria. The inner iteration stops when $\eta_i \leq \epsilon_i$ and the active-set algorithm stops when $\eta_o \leq \epsilon_o$. Also, ϵ_i is set as $0.5 \times 0.1^{k_o - 1}$ where k_o is the number of outer iterations. ϵ_o is set as 0.0001 in this simulation study. It is seen that in general the distributed active-set method can converge to a close-to-optimal solution within a few iterations. The active-set method (Algorithm 4) with fixed number of iterations converge to the optimal solution if the maximum number of the inner iterations is chosen to be large enough (e.g., 100 and 50 in this study), but the corresponding convergence time might be long (e.g., around 400 and 200, respectively). Reducing the maximum number of inner iterations to 15 speeds up the convergence but fails to

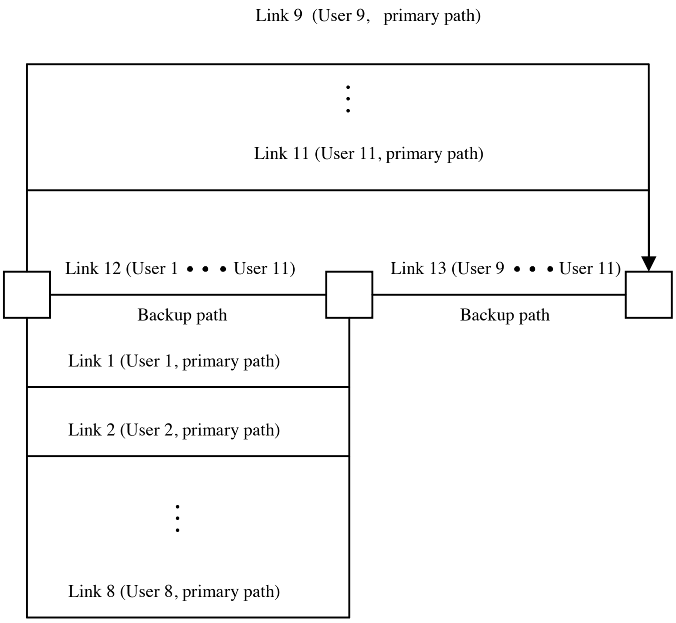

Fig. 1 Network Topology.

converge to the exact optimal solution.. In contrast, the adaptive active-set method quickly converge to the optimal solution, as shown in the last subplot.

Figure 3 compares the convergence speed of the distributed active-set method, i.e., Algorithm 4 and the cutting-plane method, i.e., Algorithm 3. To speed up the convergence, the number of inner iterations of both algorithms are chosen based on the estimated sub-optimality gap. In general, the cutting-plane method converges a bit faster than the active-set method, as shown in Figure 3. However, it requires more message-passing at each iteration as inactive constraints are not removed. Also, the number of inactive constraints typically increase with the problem size and the number of iterations, which makes the cutting-plane method impractical for large-scale networks.

As shown in Figure 4, Algorithm 4 achieves a close-to-optimal solution within only a few iteration. Notice that $D_{th} = 1 - \eta_o$ denotes a threshold for estimated sub-optimality gap of Algorithm 4. It is seen that the proposed algorithm can achieve a solution that is 99% of the optimal one within 25 iterations. The tradeoff between the sub-optimality gap and distributiveness, i.e., the number of iterations in Algorithm 4 is shown in Figure 5.

3.4 Performance-Robustness Tradeoff

Besides the worst case robustness, we can also study the performance-robustness tradeoff by using *outage probability* as the definition of robustness. Let $\boldsymbol{x}^*$ denote

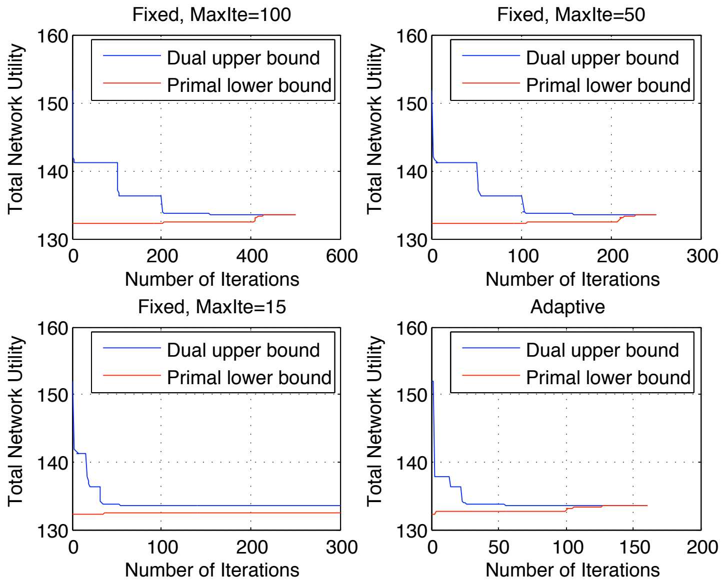

Fig. 2 Convergence behavior of the fixed and adaptive active-set method

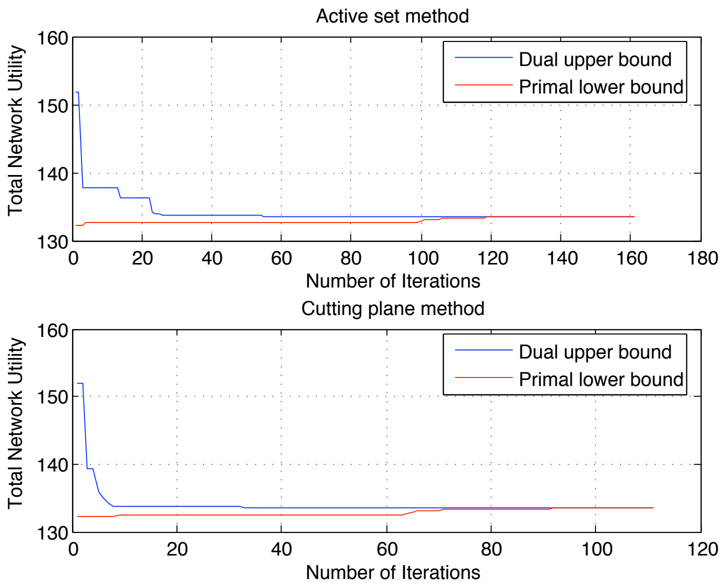

Fig. 3 Convergence behavior comparison between the active-set method and the cutting-plane method

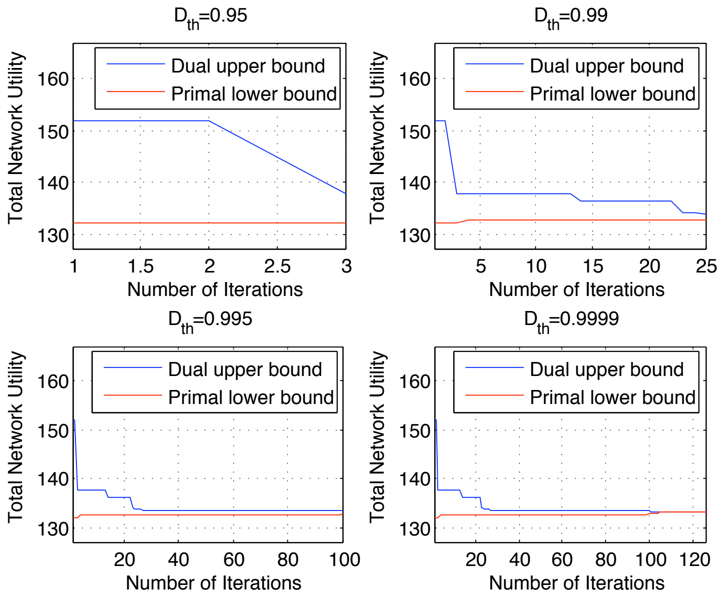

Fig. 4 Convergence behavior of algorithm 4 for different sub-optimality gaps

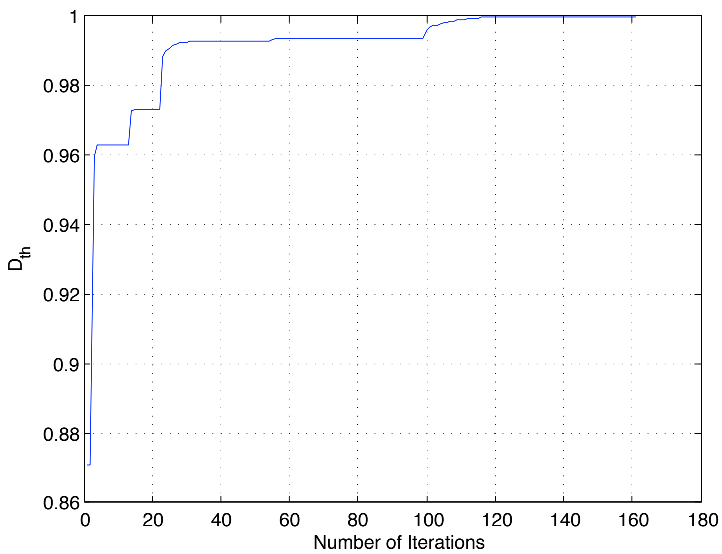

Fig. 5 Sub-optimality gap against number of iterations in algorithm 4

the optimal solution to (34). $\boldsymbol{y}^*$ is the corresponding aggregated path rates, namely, $\boldsymbol{y}^* = \boldsymbol{W}\boldsymbol{x}^*$. An outage of path t is defined as the occurrence of link failures which makes the constraint on path t is violated, i.e., $\sum_{s\in P_t} w_{ts}x_s^* + \sum_{s\in \mathcal{E}_t} \bar{b}_{ts}x_s^* > y_t^*$. Notice here $\bar{b}_{ts}$ denotes the actual percentage of rate that user s allocates to path t. Here we assume $\bar{b}_{ts}$ is a Bernoulli random number which takes value b_{ts} with probability

P_{ts} and 0 with probability $1 - P_{ts}$, i.e., $\bar{b}_{ts}$ equals to b_{ts} if the primary path fails, otherwise $\bar{b}_{ts} = 0$. The outage can be measured by a probability function $P_o(t)$ such that $P_o(t) = Pr(\sum_{s \in P_t} w_{ts} x_s^* + \sum_{s \in \mathcal{E}_t} \bar{b}_{ts} x_s^* > y_t^*)$. In addition, we use P_t to denote the upper bound to the probability of failure of the primary paths in $\mathcal{E}_t$, i.e., $P_{ts} \leq P_t$ for all $s \in \mathcal{E}_t$. We further assume failures of primary paths are independent of each other.

Clearly, the proposed rate control scheme becomes increasingly robust against link failures if we enforce stronger protections, i.e., larger values of Γ_t's. On the other hand, a larger Γ_t leads to a more conservative estimation of backup bandwidth, and consequently reduces the maximum achievable rate. Various uncertainty sets give rise to different tradeoffs between the achievable rate and outage probability. If less than Γ_t primary paths fail to work, the obtained rate control scheme remains feasible *deterministically*. Furthermore, the rate control scheme will be feasible with *high probability* if a proper Γ_t is selected, as shown in Theorem 8.

Theorem 8 *Let $\boldsymbol{x}^*$ denote the optimal solution to (34) and $\boldsymbol{y}^* = \boldsymbol{W}\boldsymbol{x}^*$. Recall $\mathcal{E}_t$ denote the set of users who utilize path t as the backup path, and $P_o(t)$ is the outage probability that the rate control on path t is violated, i.e., $Pr(\sum_{s \in P_t} w_{ts} x_s^* + \sum_{s \in \mathcal{E}_t} \bar{b}_{ts} x_s^* > y_t^*)$. Then,*

1. *$P_o(t) = 0$ if $|\bar{t}| \leq \Gamma_t$, where $|\bar{t}|$ is the number of users in $\mathcal{E}_t$ whose primary paths fail.*
2. *$P_o(t) \leq \sum_{k=\Gamma_t+1}^{|\mathcal{E}_t|} \binom{|\mathcal{E}_t|}{k} (P_t)^k (1-P_t)^{|\mathcal{E}_t|-k}$.*
3. *$P_o(t) \leq -2\frac{(\Gamma_t + 1 - |\mathcal{E}_t| P_t)^2}{|\mathcal{E}_t|}$ if $\frac{\Gamma_t+1}{|\mathcal{E}_t|} \geq P_t$. In addition, let $f_t \triangleq \frac{\Gamma_t}{\mathcal{E}_t}$. If $f_t > P_t$, we have $P_o(t) \leq e^{-D_v(f_t||p_t)|\mathcal{E}_t|}$, where $D_v(f_t||p_t)| = f_t \log\left(\frac{f_t}{P_t}\right) + (1-f_t)\log\left(\frac{1-f_t}{1-P_t}\right)$ is the Kullback-Leibler (KL) divergence for two Bernoulli random variables.*

Proof Since $|\bar{t}| \leq \Gamma_t$, we have

$$\sum_s w_{ts} x_s^* + \sum_{s \in \mathcal{F}_{t,|\bar{t}|},\ |\mathcal{F}_{t,|\bar{t}|}| = \bar{t},\ \mathcal{F}_{t,|\bar{t}|} \subseteq \mathcal{E}_t} \bar{b}_{ts} x_s^* \tag{65}$$

$$\leq \sum_s w_{ts} x_s^* + \max_{\mathcal{F}_{t,\Gamma_t}} \left(\sum_{s \in \mathcal{F}_{t,\Gamma_t},\ \mathcal{F}_{t,\Gamma_t} \subseteq \mathcal{E}_t,\ |\mathcal{F}_{t,\Gamma_t}| = \Gamma_t} \bar{b}_{ts} x_s^* \right) \tag{66}$$

$$\leq y_t^* \tag{67}$$

It follows that $P_o(t) = 0$.

Because the failures of primary paths are independent of each other, the total number of failed paths follows Binomial distribution. In addition, the failure probability of each single primary path is upper bounded by P_t. It follows that

$$P_o(t) \leq \sum_{k=\Gamma_t+1}^{|\mathcal{E}_t|} \binom{|\mathcal{E}_t|}{k} (P_t)^k (1-P_t)^{|\mathcal{E}_t|-k}. \tag{68}$$

This bound can be difficult to compute when $|\mathcal{E}_t|$ is large. We can further upper-bound the outage probability as follows.

$$P_o(t) \le \sum_{k=\Gamma_t+1}^{|\mathcal{E}_t|} \binom{|\mathcal{E}_t|}{k} (P_t)^k (1-P_t)^{|\mathcal{E}_t|-k}, \quad (69)$$

$$= \sum_{k'=0}^{|\mathcal{E}_t|-\Gamma_t-1} \binom{|\mathcal{E}_t|}{|\mathcal{E}_t|-k'} (P_t)^{|\mathcal{E}_t|-k'} (1-P_t)^{k'}, \quad (70)$$

$$= \sum_{k'=0}^{|\mathcal{E}_t|-\Gamma_t-1} \binom{|\mathcal{E}_t|}{k'} (P_t)^{|\mathcal{E}_t|-k'} (1-P_t)^{k'}, \quad (71)$$

$$\le e^{-2\frac{[|\mathcal{E}_t|(1-P_t)-(\mathcal{E}_t-\Gamma_t-1)]^2}{|\mathcal{E}_t|}}, \quad (72)$$

$$= e^{-2\frac{(\Gamma_t+1-|\mathcal{E}_t|P_t)^2}{|\mathcal{E}_t|}}, \quad (73)$$

provided that $\frac{\Gamma_t+1}{|\mathcal{E}_t|} \ge P_t$. (70) is obtained by letting $k' \triangleq |\mathcal{E}_t| - k$. (71) is due to the fact that $\binom{|\mathcal{E}_t|}{|\mathcal{E}_t|-k'}$ equals to $\binom{|\mathcal{E}_t|}{k'}$. (72) follows from the Hoeffding inequality.

Let $f_t \triangleq \frac{\Gamma_t}{\mathcal{E}_t}$. Assume $f_t > P_t$, it follows from the Chernoff-Hoeffding theorem [23] that,

$$P_o(t) \le \sum_{k=\Gamma_t+1}^{|\mathcal{E}_t|} \binom{|\mathcal{E}_t|}{k} (P_t)^k (1-P_t)^{|\mathcal{E}_t|-k}, \quad (74)$$

$$\le \left(\frac{P_t}{f_t}\right)^{f_t} \left(\frac{1-P_t}{1-f_t}\right)^{1-f_t}, \quad (75)$$

$$\le e^{-D_v(f_t||p_t)|\mathcal{E}_t|}. \quad (76)$$

Here $D_v(f_t||p_t)| = f_t \log\left(\frac{f_t}{P_t}\right) + (1-f_t)\log\left(\frac{1-f_t}{1-P_t}\right)$ denotes the Kullback-Leibler (KL) divergence for two Bernoulli random variables.

Figure 6 shows the tradeoff between robustness and performance. The performance is measured by the total network utility $\sum_s \log(x_s)$, and the robustness level is measured by the number of failures that is guaranteed to be protected on path 12, i.e., Γ_{12}. The value of Γ_{13} is fixed at 3 in this example. As we see, the performance decreases as the robustness (Γ_{12}) increases. Also the centralized algorithm and distributed Algorithm 4 achieve the same performance.

The outage probability of path 12 and its upper bounds are illustrated in Figure 7. Here P_t is set as 0.1, i.e., the probability of failures of users in $\mathcal{E}_{12}$ is upper bounded by 0.1. As evident from the figure, the outage probability decreases *exponentially* with the increase of Γ_{12}. The Chernoff and Hoeffding bounds can be used as efficient means to estimate the outage probability in case the exact outage probability is difficult to calculate.

Figure 8 illustrates the probabilistic tradeoff between robustness and total network utility. It is seen that by using the D-norm uncertainty set, the outage probability of a path can be reduced in an exponential rate by slightly decreasing the total network utility. In practice, we can select a proper Γ_{12} to strike a balance between the robustness (outage probability) and the total system throughput.

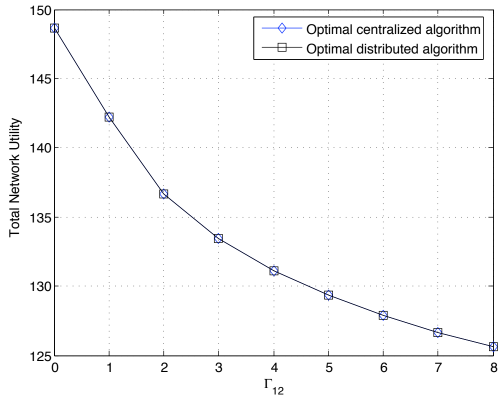

Fig. 6 Deterministic robustness-rate tradeoff under D-norm uncertainty set

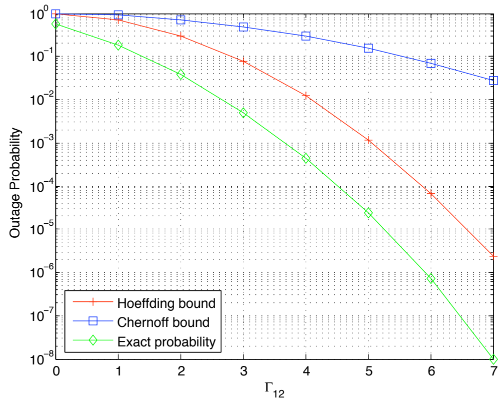

Fig. 7 Outage probability and upper bounds for different Γ_{12}

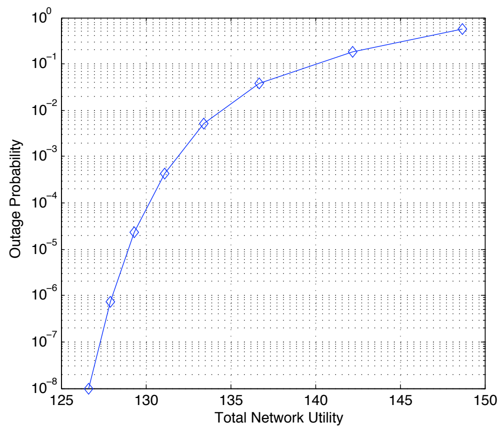

Fig. 8 Probabilistic robustness-rate tradeoff under D-norm uncertainty set

3.5 Robustness-Distributiveness Tradeoff

We next consider the tradeoff between the robustness and distributiveness of the proposed rate control algorithm. Here the robustness of the path t is quantified by the parameter Γ_t, which can control the outage probability $P_o(t)$. We use the total number of message passing required at one iteration in Algorithm 2 (Step 5 of Algorithm 4) to measure the distributiveness. At each iteration of Algorithm 2, every link will collect tentative decision rate $\boldsymbol{x}^*$ from sources (Step 4) and perform the subgradient projection as in (49). The amount of message passing required at Step 4 is *independent* of the parameter Γ_t, and the amount of message passing for Step 7 increases only linearly with $|\bar{\mathcal{G}}_l|$. So the total number of message passing during one iteration for one user is $O(|\bar{\mathcal{G}}_l|)$.

Figure 9 shows the robustness-distributiveness tradeoff of the distributed robust rate control algorithm with different thresholds, i.e., $D_{th} = 1 - \eta_o$, for estimated sub-optimality gap in Algorithm 4. The upper bound to the amount of message passing is derived from the fact that $|\bar{\mathcal{G}}_l|$ is upper bounded by $\prod_{d_{lt}=1} \binom{|\mathcal{E}_t|}{\Gamma_t}$. We also calculate the actual number of message passing at the final iteration of Algorithm 4 through simulation studies. This figure quantifies the intuition that the distributiveness of Algorithm 4 could be improved at the expense of optimality, i.e., less message-passing is required if we allow for a relatively bigger sub-optimality gap.

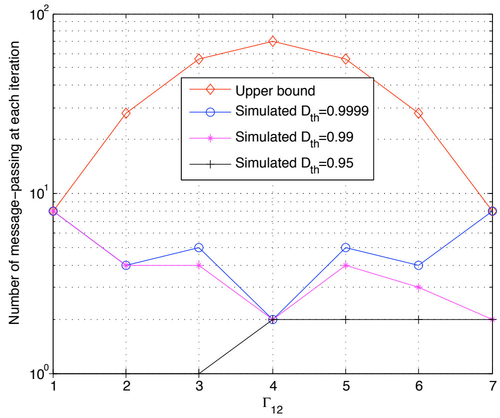

Fig. 9 Robustness-distributiveness tradeoff of the distributed active-set algorithm

3.6 Rate-Distributiveness Tradeoff with Guaranteed Outage Bound

We conclude this section with a stronger and somewhat surprising result than generally obtainable in DRO: due to the special property of the D-norm protection function in rate control, for a given path t, we can tradeoff between the total network utility and distributiveness without loss of robustness. This tradeoff is achieved by choosing different linear constraints to approximate the D-norm uncertainty set.

Recall the D-norm protection function is $g_t(\boldsymbol{b}_t, \boldsymbol{x}) = \max_{\mathcal{F}_{t,\Gamma_t} \subseteq \mathcal{E}_t} \sum_{s \in \mathcal{F}_{t,\Gamma_t}} b_{ts} x_s, \forall t$. Further, it has been seen in the preceding section that the number of required message passing at each step is proportional to $\binom{|\mathcal{E}_t|}{\Gamma_t}$, which equals to the number of linear constraints required to fully characterize $g_t(\boldsymbol{b}_t, \boldsymbol{x})$. The motivation here is to find a new protection function with a larger object function value than that of $g_t(\boldsymbol{b}_t, \boldsymbol{x})$ and can be represented by fewer linear constraints.

Consider, for example, that we want to protect the rate control scheme against single-path failure for the path t, i.e., $\Gamma_t = 1$. We assume $\mathcal{E}_t = \{e_t(1), e_t(2), e_t(3), e_t(4), e_t(5), e_t(6)\}$, where $e_t(i)$ denotes a single-path failure of the primary path of user i. The protection function is chosen to be

$$g_t(\boldsymbol{b}_t, \boldsymbol{x}) = \max_{\mathcal{F}_{t,1} \subseteq \mathcal{E}_t} \sum_{s \in \mathcal{F}_{t,1},\ |\mathcal{F}_{t,1}|=1} b_{ts} x_s. \tag{77}$$

Since $|\mathcal{E}_t| = 6$, this protection function can be equivalently represented by six linear constraints. Let $\bar{\mathcal{E}}_t = \{[e_t(1), e_t(2)], [e_t(3), e_t(4)], [e_t(5), e_t(6)]\}$. We can use the following protection function

$$\bar{g}_t(\boldsymbol{b}_t, \boldsymbol{x}) = \max_{\bar{\mathcal{F}}_t \in \bar{\mathcal{E}}_t} \sum_{s \in \bar{\mathcal{F}}_t} b_{ts} x_s, \forall t. \tag{78}$$

For any $\mathcal{F}_{t,1}$, we are able to find a corresponding set $\bar{\mathcal{F}}_t \in \bar{\mathcal{E}}_t$ such that $\mathcal{F}_{t,1} \subseteq \bar{\mathcal{F}}_t$. Further, $\bar{\mathcal{E}}_t$ contains only three sets i.e., $|\bar{\mathcal{E}}_t| = 3$. We can therefore reduce the number of required message passings at each iteration of Algorithm 4 by replacing the protection function $g_t(\boldsymbol{b}_t, \boldsymbol{x})$ with $\bar{g}_t(\boldsymbol{b}_t, \boldsymbol{x})$.

We can further obtain the following general result. Recall the D-norm protection function is given by

$$g_t(\boldsymbol{b}_t, \boldsymbol{x}) = \max_{\mathcal{F}_{t,\Gamma_t},\ |\mathcal{F}_{t,\Gamma_t} = \Gamma_t|,\ \mathcal{F}_{t,\Gamma_t} \subseteq \mathcal{E}_t} \sum_{s \in \mathcal{F}_{t,\Gamma_t}} b_{ts} x_s, \forall t, \tag{79}$$

where $\mathcal{E}_t = \{s : b_{ts} > 0, \forall s\}$ is the set of users who utilize path t as the backup path. $\mathcal{F}_{t,\Gamma_t}$ denotes a subset of $\mathcal{E}_t$ with size Γ_t. Let $\bar{g}_t(\boldsymbol{b}_t, \boldsymbol{x})$ denote another protection function such that

$$\bar{g}_t(\boldsymbol{b}_t, \boldsymbol{x}) = \max_{\bar{\mathcal{F}}_t \in \bar{\mathcal{E}}_t} \sum_{s \in \bar{\mathcal{F}}_t} b_{ts} x_s, \forall t. \tag{80}$$

Lemma 1 *Assume there exists a set $\bar{\mathcal{E}}_t$ such that for any $\mathcal{F}_{t,\Gamma_t}$, we can always find $\bar{\mathcal{F}}_t \in \bar{\mathcal{E}}_t$ which satisfies $\mathcal{F}_{t,\Gamma_t} \subseteq \bar{\mathcal{F}}_t$. The outage probability of path t by using the protection function $\bar{g}_t(\boldsymbol{b}_t, \boldsymbol{x})$ is no larger than that by using the D-norm protection function.*

Proof Let $\boldsymbol{x}^*$ denote the optimal solution to (34). $\boldsymbol{y}^*$ is the corresponding aggregated path rates, namely, $\boldsymbol{y}^* = \boldsymbol{W}\boldsymbol{x}^*$.

Assume $\{\bar{\boldsymbol{x}}\}$ is the optimal solution to the robust rate control problem (34) where the protection function is set as $\bar{g}_t(\boldsymbol{b}_t, \boldsymbol{x})$. $\bar{\boldsymbol{y}}$ represents the aggregated path rates,i.e., $\boldsymbol{y}^* = \boldsymbol{W}\boldsymbol{x}^*$. For any $\mathcal{F}_{t,\Gamma_t}$, there exist $\bar{\mathcal{F}}_t$ such that $\mathcal{F}_{t,\Gamma_t} \subseteq \bar{\mathcal{F}}_t$, it follows that

$$g_t(\boldsymbol{b}_t, \bar{\boldsymbol{x}}) = \max_{\mathcal{F}_{t,\Gamma_t} \subseteq \mathcal{E}_t} \sum_{s \in \mathcal{F}_{t,\Gamma_t}} b_{ts} \bar{x}_s \leq \max_{\bar{\mathcal{F}}_t \subseteq \bar{\mathcal{E}}_t} \sum_{s \in \bar{\mathcal{F}}_t} b_{ts} \bar{x}_s = \bar{g}_t(\boldsymbol{b}_t, \bar{\boldsymbol{x}}).$$

Consequently, we have $\sum_s w_{ts} \bar{x}_s + \bar{g}_t(\boldsymbol{b}_t, \bar{\boldsymbol{x}}) \leq \sum_s w_{ts} \bar{x}_s + g_t(\boldsymbol{b}_t, \bar{\boldsymbol{x}}) \leq \bar{y}_t$, which means $\{\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}\}$ is a feasible solution to the robust power control problem with D-norm protection function. This directly proves the claim.

The tradeoff between the total network utility and the robustness is shown in Figure 10. Here single-path protection is assumed, i.e., $\Gamma_{12} = 1$. To make sure the rate control scheme is robust against single-path failure, we can choose the D-norm protection function with $\Gamma_t = 1$. However, such a protection function brings about eight additional linear constraints and requires considerable message passing. To reduce the number of message passing, we can use an alternative protection function. Here we consider three other protection functions satisfying conditions given in Theorem 1. Since $|\mathcal{E}_{12}| = 8$. The first protection function is to let $|\bar{F}_{12}| = 2$ and $\bar{\mathcal{E}}_t = \{[e_t(1), e_t(2)], [e_t(3), e_t(4)], [e_t(5), e_t(6)], [e_t(5), e_t(6)]\}$. Such a protection function can be represented by four linear constraints. We can obtain the other two protection functions in a similar manner.

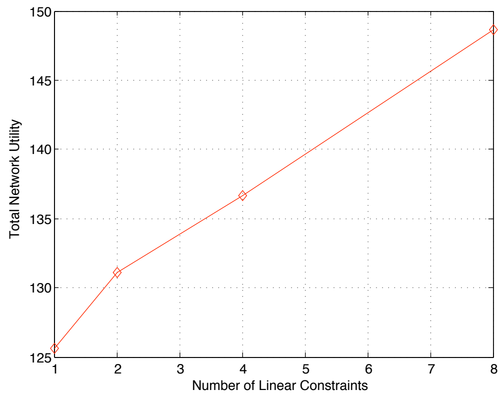

Fig. 10 Rate-distributiveness tradeoff with guaranteed outage bound

4 Conclusions

Making communication network optimization models robust and distributed at the same time is an under-explored area. This paper initiates the study of DRO through several robust formulations that preserve a large degree of distributiveness of solution algorithms. We first describe several models for describing parameter uncertainty sets that can lead to distributed solutions for linearly constrained nominal problems. These models include general polyhedron, D-norm, and ellipsoid. We then apply these models in the example of distributed rate control. The tradeoff between robustness (i.e., the maximum of link failures allowed), performance (total network utility), and distributiveness (i.e., the amount of message passing needed) is demonstrated. In Part II of this two-part paper, extensive applications of DRO methodology to wireless power control will be further presented.

The study of distributed robust optimization in general remains open, with many challenging issues and possible applications where robustness to uncertainty is as important as optimality in the nominal model. In particular, possible extensions include nonlinear constraint sets in the nominal problem or uncertainties in other parts of the nominal formulations.

References

1. A. Ben-Tal and A. Nemirovski, “Robust solutions to uncertain programs,” *Operations Research Letter*, vol. 25, pp. 1–13, Feb. 1999.
2. ——, “Selected topics in robust convex optimization,” *Mathematical Programming*, vol. 1, no. 1, pp. 125–158, July 2007.
3. A. Nemirovski, “On tractable approximations of randomly perturbed convext constaints,” *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 2419–2422, 2003.
4. L. El-Ghaoui and H. Lebret, “Robust solutions to least-square problems to uncertain data matrices,” *SIAM J. Matrix Anal. Appl.*, vol. 18, pp. 1035–1064, 1997.
5. L. El-Ghaoui, F. Oustry, and H. Lebret, “Robust solutions to uncertain semidefinite programs,” *SIAM J. Optim.*, vol. 9, pp. 33–52, 1998.
6. A. Ben-Tal and A. Nemirovski, “Robust solutions to linear programming problems contaminated with uncertain data,” *Mathematical Programming*, vol. 88, pp. 411–424, 2000.
7. D. P. Bertsekas and J. Tsitsikklis, *Parallel and distributed computation: numerical methods*. Englewood Cliffs, NJ: Prentice Hall, 1999.
8. G. Foschini and Z. Miljanic, “A simple distributed autonomous power control algorithm and its convergence,” *Vehicular Technology, IEEE Transactions on*, vol. 42, no. 4, pp. 641–646, Nov. 1993.
9. J. Huang, R. Berry, and M. L. Honig, “Distributed interference compensation in wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 5, pp. 1074–1084, May 2006.
10. J. W. Lee, M. Chiang, and R. A. Calderbank, “Utility-optimal random-access control,” *Wireless Communications, IEEE Transactions on*, vol. 6, no. 7, pp. 2741–2751, 2007.
11. F. Kelly, A. Maulloo, and D. Tan, “Rate control for communication networks: shadow prices, proportional fairness and stability,” *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.
12. K. Yang and X. Wang, “Cross-layer network planning for multi-radio cognitive wireless networks,” *Communications, IEEE Transactions on*, vol. 7, no. 11, pp. 1363–1373, November 2008.
13. B. Dimitris and M. Sim, “The price of robustness,” *Operations Research*, vol. 52, no. 1, pp. 35–53, Feb. 2004.
14. S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
15. A. Ben-Tal and A. Nemirovski, “Robust solutions to uncertain linear problems,” Research Report 6/95, Optimization Laboratory, Faculty of Industrial Engineering and Management, Technion, Tech. Rep., 1995.
16. D. Bertsimas and J. N. Tsitsiklis, *Linear Optimization*, 1st ed. Belmont, MA: Athena Scientific, 1997.
17. M. Giaquinta and S. Hildebrandt, *Calculus of variations*, 1st ed. New York: Springe, 1996.
18. A. N. A. Ben-Tal, “On polyhedral approximations of the second-order cone,” *Mathematics of Operations Research*, vol. 26, no. 2, pp. 193–205, May 2001.
19. A. Yener, R. Yates, and S. Ulukus, “CDMA multiuser detection: A nonlinear programming approach,” *IEEE Transactions on Communications*, vol. 50, no. 6, pp. 1016–1024, June 2002.
20. K. Yang, J. Feldman, and X. Wang, “Non-linear programming approaches to decoding low-density parity-check codes,” *IEEE J. Select. Areas Commun.*, vol. 24, no. 6, pp. 1603–1613, Aug. 2006.
21. D. Xu, Y. Li, M. Chiang, and A. R. Calderbank, “Optimal provision of elastic service availability,” *Proc. IEEE INFOCOM*, 2007.
22. S. Leyffer, “The return of the active set method,” *Oberwolfach Reports*, vol. 2, no. 1, pp. 107–109, January 2005.
23. W. Hoeffding, “Probability inequalities for sums of bounded random variables,” *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, March 1963.