

SOLUTION APPROACHES TO LINEAR FRACTIONAL PROGRAMMING AND ITS STOCHASTIC GENERALIZATIONS USING SECOND ORDER CONE APPROXIMATIONS *

CHEOLMIN KIM [†] AND SANJAY MEHROTRA [‡]

Abstract. We consider linear fractional programming problems in a form of which the linear fractional program and its stochastic and distributionally robust counterparts with finite support are special cases. We introduce a novel reformulation that involves differences of square terms in the constraint, subsequently using a piecewise linear approximation for the concave part. Using the resulting second order cone programs (SOCPs), we develop a solution algorithm in the branch and bound framework. Our method iteratively refines the piecewise linear approximations by dividing hyper-rectangles and solves SOCPs to obtain lower bounds for the sub-hyper-rectangles. We derive a bound on the optimality gap as a function of the approximation errors at the iterate and prove that the number of iterations to attain an ϵ -optimal solution is in the order of $\mathcal{O}(\sqrt{\epsilon})$. Numerical experiments show that the proposed algorithm scales better than state-of-the-art linear-programming-based algorithms and commercial solvers to solve linear fractional programs. Specifically, the proposed algorithm achieves two or more digits of accuracy in significantly less time than the time required by the known algorithms on medium to larger size problem instances. Experimental results with Wasserstein ambiguity sets reveal that our reformulation-based approach solves small size distributionally robust linear fractional programs, with the cardinality of support up to 25.

Key words. Distributionally robust optimization, linear fractional programming, second order cone approximations, branch and bound algorithm, convergence analysis

AMS subject classifications. 90C26, 90C32, 90C57

1. Introduction. We study the linear fractional programming problem in the form

$$\begin{aligned} \vartheta^* := \min \quad & \theta \\ \text{subject to} \quad & \frac{c_k^T x + \alpha_k}{d_k^T x + \beta_k} \leq \gamma_k, \quad k \in [K] := \{1, \dots, K\}, \\ & f^T \pi \leq \theta, \quad H^T \pi \geq \gamma, \quad P\gamma \leq \theta \mathbf{1}_J, \\ & x \in \mathcal{X}, \quad \theta \in \mathbb{R}, \quad \gamma \in \mathbb{R}^K, \quad \pi \in \mathbb{R}^L, \end{aligned} \tag{1.1}$$

where

(A1) $\mathcal{X} := \{x \mid Ax \leq b, x \geq 0\}$ is a non-empty polytope in $\mathbb{R}^n$.

(A2) $c_k^T x + \beta_k \geq 0$ for all $x \in \mathcal{X}$ and $k \in [K]$.

(A3) $d_k^T x + \beta_k > 0$ for all $x \in \mathcal{X}$ and $k \in [K]$.

(A4) P is a non-negative matrix in $\mathbb{R}^{J \times K}$.

(A5) $\mathcal{P} := \{p \mid Hp = f, p \geq 0\}$ is a non-empty polytope in $\mathbb{R}^K$.

The assumptions (A1)-(A3) are commonly made in the literature of linear fractional programming (19; 23; 20; 2; 35; 24; 26). Assumptions (A4)-(A5) are introduced to develop solution approaches for stochastic and distributionally robust linear fractional programs. In this context, the set $\mathcal{P}$ represents a polyhedral ambiguity set (see Section 6.1). However, in the following discussion we

*Submitted to the editors December 19, 2019.

Funding: This work was funded by the National Science Foundation under grant CMMI 1763935.

[†]Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL (cheolmkim@u.northwestern.edu).

[‡]Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL (mehrotra@northwestern.edu).

33 take it as a general set unless otherwise specified. By (A1)-(A5), the optimization problem (1.1)
 34 has a finite optimum value ϑ^* .

35 Model (1.1) covers the following linear fractional problems in its framework:

36 • (Stochastic) Linear Fractional Program: The linear fractional program can be written as

$$\begin{aligned} \vartheta^* := \min \quad & \theta \\ \text{subject to} \quad & \frac{c_k^T x + \alpha_k}{d_k^T x + \beta_k} \leq \gamma_k, \quad k \in [K], \\ & p^T \gamma \leq \theta, \quad x \in \mathcal{X}, \quad \theta \in \mathbb{R}, \quad \gamma \in \mathbb{R}^K, \end{aligned} \quad (1.2)$$

38 which has $|J| = 1$ and does not include π variables and their constraints $f^T \pi \leq \theta$, $H^T \pi \geq \gamma$
 39 in (1.1). If the vector $p \in \mathbb{R}_+^K$ satisfies $\sum_{k=1}^K p_k = 1$, the formulation (1.2) reduces to the
 40 stochastic linear fractional program with finite support or a sample average approximation
 41 (SAA) to the stochastic linear fractional program of the form:

$$(1.3) \quad \min_{x \in \mathcal{X}} \mathbb{E}_{\mathbb{P}} \left[\frac{\tilde{c}^T x + \tilde{\alpha}}{\tilde{d}^T x + \tilde{\beta}} \right].$$

43 In the above, $\mathbb{P}$ is a probability distribution governing the random vector $\tilde{\xi} = (\tilde{c}, \tilde{d}, \tilde{\alpha}, \tilde{\beta})$.
 44 The (stochastic) linear fractional problem (1.2) arises in various decision problems such as
 45 multi-stage shipping (1), cluster analysis (38), and multi-objective bond portfolio optimiza-
 46 tion (22), to name a few. For additional applications, see (39; 45).

47 • Distributionally Robust Linear Fractional Program (Section 6): While *stochastic optimization*
 48 minimizes the expected value of an objective function with respect to a *known* proba-
 49 bility distribution $\mathbb{P}$, *distributionally robust optimization* (DRO) seeks a solution that min-
 50 imizes the worst-case expected value over an *ambiguity set* $\mathcal{U}$ of probability distributions $\mathbb{P}$
 51 as

$$(1.4) \quad \min_{x \in \mathcal{X}} \sup_{\mathbb{P} \in \mathcal{U}} \mathbb{E}_{\mathbb{P}} \left[\frac{\tilde{c}^T x + \tilde{\alpha}}{\tilde{d}^T x + \tilde{\beta}} \right].$$

53 For polyhedral and convex ambiguity sets, we can write a reformulation or subproblem of
 54 (1.4) in the form of (1.1) as follows:

- 55 – For a polyhedral ambiguity set $\mathcal{P}$, using the linear programming duality, the distribu-
 56 tionally robust linear fractional program with finite support can be reformulated to
 57 (1.1) without the constraint $P\gamma \leq \theta \mathbf{1}_J$ as seen in (6.2). The constraints $Hp = f$ and
 58 $p \geq 0$ in $\mathcal{P}$ are dualized as $f^T \pi \leq \theta$ and $H^T \pi \geq \gamma$.
- 59 – For a convex ambiguity set $\mathcal{C}$, the distributionally robust linear fractional program with
 60 finite support can be solved by a cutting surface algorithm that iteratively solves a
 61 subproblem (6.12) of the form (1.1) without π variables and their constraints $f^T \pi \leq \theta$
 62 and $H^T \pi \geq \gamma$. In (6.12), each row of P belongs to the ambiguity set $\mathcal{C}$.

63 **1.1. Contributions.** We introduce the linear fractional model (1.1) and develop an algorithm
 64 for its solution. Our model includes the linear fractional program and its stochastic and distribu-
 65 tionally robust counterparts with finite support in a common framework. In order to solve this
 66 model, we introduce a non-convex quadratic reformulation using difference of squares. Existing
 67 non-convex quadratic reformulations in the literature use bilinear terms and their McCormick re-
 68 laxations. In our approach applying a piecewise linear approximation to concave quadratic terms

yields a mixed binary second order cone program (MB-SOCP). Instead of solving a sequence of computationally expensive MB-SOCP problems, we develop a branch and bound algorithm that adaptively refines the piecewise linear approximations and iteratively solves SOCP problems instead of MB-SOCP problems.

Our spatial branch and bound algorithm works in the space of the variables that appear in the nonconvex quadratic terms. It successively bisects a hyper-rectangle and solves an SOCP approximation problem for each sub-hyper-rectangle to obtain a lower bound. The algorithm iteratively updates the incumbent solution and the lower bound until the relative optimality gap becomes smaller than a given tolerance level ϵ . In the convergence analysis, we prove a bound on the optimality gap as a function of approximation errors at the iterate. Using this bound, we show that our branch and bound algorithm attains an ϵ -optimal solution in a finite number of iterations. Particularly, we give an upper bound for the number of iterations, which has been not done in previous works in the literature where convergence analyses are based on limiting arguments. Due to the second order cone approximations, the worst-case bound is in the order of $\mathcal{O}(\sqrt{\epsilon})$, which demonstrates the efficiency of the proposed algorithm in finding an accurate solution.

We report numerical experiments comparing the performance of our algorithm with GloMIQO (33) and Gurobi commercial solvers. These solvers were used to solve a non-convex quadratic formulation of the problem. Comparisons are also made with several state-of-the-art algorithms (19; 26; 35) which employ linear programming (LP) approximations within the branch and bound framework. Our results show that the proposed algorithm scales better than both the LP-based algorithms and the off-the-shelf solvers with the size of the problems. The proposed algorithm achieves two or more digits of accuracy faster than the benchmark algorithms on medium and large size problems. Even for small size problems, our algorithm attains five digits of accuracy fastest in most cases, which can be attributed to its square root dependency on ϵ . This improvement grows with problem dimensionality and the number of terms in the linear fractional program. For example, the proposed method managed to achieve five-digit of accuracy for problem with $(n, K) = (100, 10)$ in about 40 minutes on average; where the best known algorithm achieved 2-digit accuracy in nearly 6 hours on average.

We introduce solution approaches for distributionally robust linear fractional programs (1.4) with finite support. For a polyhedral ambiguity set, we derive an equivalent formulation of the form (1.1) using the LP duality. We provide three examples of such reformulations for popular polyhedral ambiguity sets based on monomial moments, total variation distance, and the Wasserstein metric. For a general convex ambiguity set, we introduce a cutting surface algorithm where we iteratively solve a subproblem of the form (1.1) together with an LP separation problem which results in a probability cut. The experiments using Wasserstein ambiguity sets indicate that the reformulation approach solves small size instances with the cardinality of support up to 25.

1.2. Organization of this Paper. This paper is organized as follows. We provide a literature review on linear fractional programming and distributionally robust optimization in Section 2. We introduce second order cone approximations in Section 3 and develop an adaptive branch and bound algorithm in Section 4. We give a convergence analysis in Section 5. Distributionally robust counterparts are discussed in Section 6 and the experimental results are discussed in Section 7.

2. Literature Review. In this section, we provide the current literature on linear fractional programming and a brief review on distributionally robust optimization.

2.1. Linear Fractional Programming. The literature review on linear fractional programming consists of two parts. The first part considers some special cases of (1.2) where K is small

or the objective function f has a special structure in x . In the second part, we review algorithms based on the outer approximation or the branch and bound approach.

2.1.1. Special Cases.

A Single Linear Fractional Function Case. A single linear fractional function is pseudo-linear. Therefore, an optimal solution is an extreme point of $\mathcal{X}$ and all local optima are globally optimal (25). Exploiting these properties, various solution approaches such as simplex algorithm (47), parametric method (11), geometric approach (44), change of variable (9) and gradient-based method (15) have been developed (45). Among them, the Charnes-Cooper transformation (9) gives a reformulation of the problem to an LP, making it easily solvable using an off-the-shelf solver.

Sum of Two Linear Fractional Functions Case. If $K > 1$, due to the coupling of linear fractional functions with respect to x , the Charnes-Cooper transformation technique does not yield an LP or a convex optimization problem. Also, neither pseudo-concavity nor pseudo-convexity is preserved under summation of linear fractional functions. Therefore, an optimal solution is not necessarily at a vertex and multiple local optimal solutions can exist, making it challenging to solve (1.2). Nonetheless, if $K = 2$, an optimal solution belongs to an edge of $\mathcal{X}$ (8). Based on this property, simplex-type algorithms using exhaustive search (8; 23) have been developed to find an exact global optimum of (1.2). Applying the Charnes-Cooper transformation (9) to one linear fractional function, these works consider an equivalent formulation minimizing the sum of a linear and a linear fractional function under a polytope. Fixing the value of the denominator of the remaining linear fractional function, they obtain a parametric linear program. Parametric simplex algorithms (8; 23) generate a sequence of bases associated with an optimal basic solution of the parametric linear program with varying the value of the parameter until they find the one that results in a global minimum. No generalization of this approach is known for the case $K > 2$. However, a heuristic algorithm (20) combining a grid search for one linear fractional function and the exact algorithm (23) for the other two linear fractional functions has been developed for $K = 3$.

Special Cases of Sum of More than Two Linear Fractional Functions. Assuming that $d_k \geq 0$, i.e., it is non-negative componentwise, and $\beta_k = 1$ for $k \in [K]$, an equivalent formulation minimizing $\sum_{k=1}^K (c_k - f_k d_k)^T x$ subject to $x \in \mathcal{X}$, $f_k \geq (c_k - f_k d_k)^T x + a_k$, $k \in [K]$ is considered in (2). This work extends the parametric approach for a general single fractional program introduced in (11). If the objective function is separable in x , meaning that the fractional terms are defined for disjoint subsets of x , a vector f^* satisfying local optimality conditions is unique and every local optimal solution is globally optimal. To find such a vector f^* , a gradient-based algorithm updating f at each iteration is presented in (2) with convergence guarantees for $K \leq 3$ under the separability assumption.

2.1.2. The General Case.

Outer Approximation Approaches. Introducing two auxiliary variables u_k and v_k for each linear fractional term, (24; 4) consider equivalent optimization problems that minimize a concave function subject to a convex set in the space of u and v . Using the fact that a concave minimization problem under a polyhedron has an optimal solution at an extreme point of the polyhedron, (24) develops an outer approximation algorithm that solves a sequence of concave minimization problems under a polyhedron. Iteratively refining the polyhedron by adding cutting planes, this algorithm attains an ϵ -optimal solution in a finite number of steps. On the other hand, outer approximation algorithms in (13; 35) consider the image space $S = \{\gamma \in \mathbb{R}^K \mid \gamma_k = (c_k^T x + \alpha_k)/(d_k^T x + \beta_k), k \in [K], x \in \mathcal{X}\}$. To minimize $\sum_{k=1}^K \gamma_k$ subject to $\gamma \in S$, (13) proposes an algorithm that iteratively shrinks the simplex containing an optimal solution γ^* . The extreme points of the simplex are up-

dated one at a time by solving a series of LP problems. However, this approach has no convergence analysis. Another outer approximation approach using a union of boxes which contains S is introduced in (35). The union of boxes is called a polyblock. Based on the observation that an optimal solution $\hat{\gamma}$ to an approximation problem occurs at an extreme point of the polyblock, this algorithm maintains all extreme points of the polyblock and returns the one having the best objective value. If $\hat{\gamma} \in S$, $\hat{\gamma}$ is optimal. Otherwise, $\hat{\gamma}$ is projected to S to produce a new set of extreme points which replace $\hat{\gamma}$ in the polyblock. Unfortunately, the projection step is as hard as the original problem. Therefore, an approximate projection step, which involves a sequence of LP problems, is used in a practical development.

Branch and Bound Approaches. Using K auxiliary variables γ_k , (21) considers an equivalent non-convex quadratically constrained optimization program that minimizes $\sum_{i=1}^K \gamma_k$ subject to $x \in \mathcal{X}$ and $c_k^T x + \alpha_k - \gamma_k(d_k^T x + \beta_k) \leq 0$ where $k \in [K]$. To handle these bilinear terms in the constraint, (21) uses McCormick envelopes (31) and, as a result, it obtains an LP relaxation problem. By iteratively branching a hyper-rectangle in the space of (x, γ) and solving LP relaxation problems for each sub-hyper-rectangle, this algorithm attains an ϵ -optimal solution in a finite number of steps. On the other hand, (26; 19) consider equivalent formulations with non-convex objective functions. Using additional variables s_k and t_k for each fractional function, (19) considers an equivalent formulation that minimizes $\sum_{k=1}^K s_k t_k$ subject to $x \in \mathcal{X}$, $c_k^T x + \alpha_k \leq t_k$ and $s_k(d_k^T x + \beta_k) \geq 1$ where $k \in [K]$. Using McCormick envelopes (31) for the bilinear terms in the objective function and replacing s_k in the constraint with its upper bound, it derives an LP approximation problem. Branching on s_k , this LP approximation problem is iteratively solved in the branch and bound algorithm. Introducing extra variables ξ_k and η_k for each fractional term, (26) studies an equivalent formulation that maximizes $\sum_{k=1}^K \eta_k / \xi_k$ subject to $x \in \mathcal{X}$, $c_k^T x + \alpha_k = \eta_k$ and $d_k^T x + \beta_k = \xi_k$ where $k \in [K]$. Given lower and upper bounds of $\eta_k + \xi_k$ and η_k / ξ_k , an overestimator of η_k / ξ_k that has a similar form to McCormick envelopes (31) is developed to produce an LP approximation problem. Using this LP approximation problem, it derives a branch and bound algorithm that performs branching on η_k / ξ_k . While existing reformulations discussed above have bivariate non-convex terms such as a product or ratio of two different variables, the non-convex parts in our formulation are univariate, negative square terms. Therefore, using the same number of branching variables, more efficient branch and bound algorithm can be developed through our formulation.

2.2. Distributionally Robust Optimization (DRO). Many real-world decision problems have parameter uncertainty. To deal with parameter uncertainty, stochastic optimization models it through a probability distribution and minimizes the expected cost under the assumed distribution. On the other hand, robust optimization approach models the uncertainty by specifying a support of parameters and minimizes the worst-case cost over the set of possible values. Taking an intermediate framework between stochastic optimization and robust optimization, distributionally robust optimization models the ambiguity in parameter distribution through a set of probability distributions and minimizes the worst-case average cost over the ambiguity set of probability distributions. To develop a tractable DRO model with good performance, various ambiguity sets based on moments (5; 10; 6; 42; 41), total variation distance (40; 18; 36), ϕ -divergence (3; 18; 27; 48; 7), and the Wasserstein metric (14; 12; 30) have been introduced. To solve DRO models, most works (49; 5; 10) use the dualization of inner problems. However, if regularity conditions are not satisfied, strong duality might not hold. On the contrary, cutting surface algorithms (32; 30; 17) do not require such conditions and thus are applicable in a general setting. For a recent comprehensive

review on distributionally robust optimization, see (37).

3. Second Order Cone Approximations. In this section, we first present a reformulation of linear fractional functions to non-convex quadratic constraints. Then, we introduce the idea of using piecewise linear approximations of concave parts to obtain second order cone approximation problems.

3.1. Reformulations. The formulation (1.1) has the non-convex constraints

$$(3.1) \quad \frac{c_k^T x + \alpha_k}{d_k^T x + \beta_k} \leq \gamma_k, \quad k \in [K].$$

Using (A3), we equivalently write (3.1) as

$$(3.2) \quad c_k^T x + \alpha_k \leq (d_k^T x + \beta_k) \gamma_k, \quad k \in [K].$$

Let

$$(3.3) \quad w_k := \frac{\gamma_k + (d_k^T x + \beta_k)}{2}, \quad v_k := \frac{\gamma_k - (d_k^T x + \beta_k)}{2}, \quad k \in [K].$$

Since $(d_k^T x + \beta_k) \gamma_k = w_k^2 - v_k^2$, we represent (3.2) as the following system of constraints:

$$(3.4) \quad c_k^T x + \alpha_k + v_k^2 \leq w_k^2, \quad \gamma_k - (d_k^T x + \beta_k) = 2v_k, \quad \gamma_k + (d_k^T x + \beta_k) = 2w_k, \quad k \in [K].$$

Let

$$(3.5) \quad \mathcal{S} = \left\{ (x, \theta, \gamma, \pi, w, v) \left| \begin{array}{l} \gamma_k - (d_k^T x + \beta_k) = 2v_k, \quad \gamma_k + (d_k^T x + \beta_k) = 2w_k, \quad k \in [K], \\ f^T \pi \leq \theta, \quad H^T \pi \geq \gamma, \quad P\gamma \leq \theta \mathbf{1}_J, \\ x \in \mathcal{X}, \quad \theta \in \mathbb{R}, \quad \gamma \in \mathbb{R}^K, \quad \pi \in \mathbb{R}^L, \quad w \in \mathbb{R}^K, \quad v \in \mathbb{R}^K. \end{array} \right. \right\}.$$

Then, we can write (1.1) as

$$(3.6) \quad \begin{aligned} \vartheta^* := \min \quad & \theta \\ \text{subject to} \quad & c_k^T x + \alpha_k + v_k^2 \leq w_k^2, \quad k \in [K], \\ & (x, \theta, \gamma, \pi, w, v) \in \mathcal{S}. \end{aligned}$$

PROPOSITION 3.1. *Two optimization problems (1.1) and (3.6) are equivalent:*

- *If $(x^*, \theta^*, \pi^*, \gamma^*, w^*, v^*)$ is an optimal solution to (3.6), then the solution $(x^*, \theta^*, \pi^*, \gamma^*)$ is an optimal solution to (1.1).*
- *If $(x^*, \theta^*, \pi^*, \gamma^*)$ is an optimal solution to (1.1), then the solution $(x^*, \theta^*, \pi^*, \bar{\gamma}, \bar{w}, \bar{v})$ such that*

$$(3.7) \quad \bar{\gamma}_k = \frac{c_k^T x^* + \alpha_k}{d_k^T x^* + \beta_k}, \quad \bar{w}_k = \frac{\bar{\gamma}_k + d_k^T x^* + \beta_k}{2}, \quad \bar{v}_k = \frac{\bar{\gamma}_k - (d_k^T x^* + \beta_k)}{2}, \quad k \in [K]$$

is an optimal solution to (3.6).

Proof. See Appendix A.1. □

231 Note that the optimization problem (3.6) has all linear expressions except for the K non-convex
 232 quadratic constraints of the form

$$233 \quad (3.7) \quad c_k^T x + \alpha_k + v_k^2 \leq w_k^2, \quad k \in [K].$$

234 Formulation (3.6) is a difference-of-convex (DC) program (16). As compared to the existing reformulations in the literature that involve bilinear or bivariate terms, non-convex parts of (3.6) are
 235 difference of two square terms (3.7). To address the concave parts, one can use the convex-concave
 236 procedure (50) in DC programming, which iteratively solves convex approximation problems obtained by linearizing concave parts of the objective function and constraints. Using the solutions
 237 from the approximation problems, CCP performs a line search to update the iterates. However,
 238 since this approach repeatedly considers local approximations near the current iterates, there is no
 239 guarantee that CCP will yield a solution with the desired tolerance. Indeed, the convergence of
 240 this method is shown to a critical point (43) only. On the contrary, our approach seeks a global
 241 optimal solution based on a piecewise linear approximation of w_k^2 as explained below.

244 **3.2. Piecewise Linear Approximations.** By (A1) and (A3), $d_k^T x + \beta_k$ and $(c_k^T x + \alpha_k)/(d_k^T x + \beta_k)$
 245 are bounded for all $k \in [K]$. Also, since (1.2) has a finite optimum ϑ^* by (A1)-(A5), from
 246 Proposition 3.1, there exists an optimal solution $(x^*, \theta^*, \pi^*, \gamma^*, w^*, v^*)$ to (3.6) which satisfies

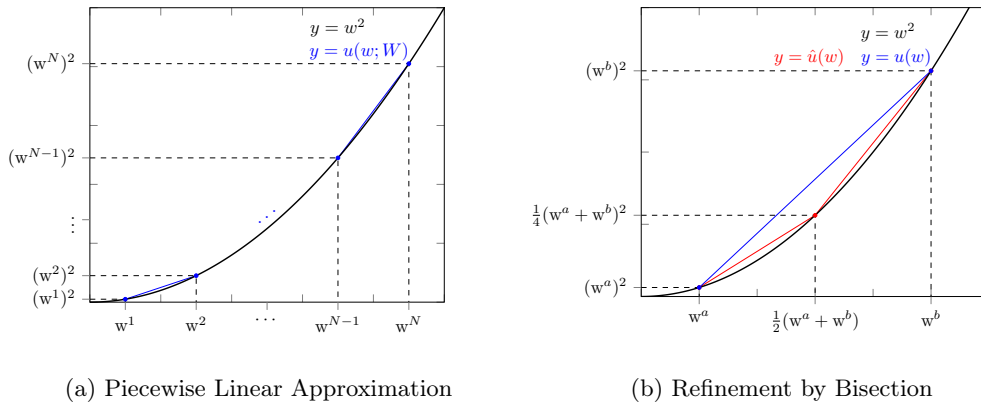
$$247 \quad (3.8) \quad \gamma_k^* = \frac{c_k^T x^* + \alpha_k}{d_k^T x^* + \beta_k}, \quad k \in [K].$$

248 Therefore, without loss of generality, we can set lower and upper bounds for γ_k , and thus for w_k
 249 due to (3.3). Let w_k^m and w_k^M be lower and upper bounds of w_k and $W_k := \{w_k^1, \dots, w_k^{N_k}\}$ be a set
 250 of points such that $w_k^m = w_k^1 \leq \dots \leq w_k^{N_k} = w_k^M$.

251 For notational convenience, we temporarily drop the subscript k . To develop an approximation
 252 of w^2 in $[w^m, w^M]$, we define a piecewise linear function u as

$$253 \quad (3.9) \quad u(w; W) = \max_{1 \leq j \leq N-1} (w^j + w^{j+1})w - w^j w^{j+1}.$$

Fig. 1: Piecewise Linear Approximation and Refinement by Bisection



As illustrated in Figure 1a, the piecewise linear function u satisfies $w^2 \leq u(w; W)$ for every $w \in [w^m, w^M]$. The approximation error at w defined as $u(w; W) - w^2$ depends on the two neighboring points $w^a = \max_{w \in W, w \leq w} w$ and $w^b = \min_{w \in W, w \leq w} w$. If $w = w^a$ or w^b , the approximation error at w is zero. Otherwise, it depends on the magnitude of $w^b - w^a$ and the location of w in $[w^a, w^b]$. The approximation error at $w \in [w^a, w^b]$ remains the same if we add a point outside of $[w^a, w^b]$ to W but decreases if a point in $[w^a, w^b]$ is added to W . The following proposition states that the worst-case approximation error has a quadratic relation with $w^b - w^a$ and adding the midpoint $(w^a + w^b)/2$ to W decreases the approximation error at least by half for all $w \in [w^a, w^b]$ (see Figure 1b).

PROPOSITION 3.2. Let $u(w) = (w^a + w^b)(w - w^a) + (w^a)^2$ and

$$\hat{u}(w) = \begin{cases} \frac{1}{2}((3w^a + w^b)w - w^a(w^a + w^b)), & w \in \left[w^a, \frac{w^a + w^b}{2}\right] \\ \frac{1}{2}((w^a + 3w^b)w - (w^a + w^b)w^b), & w \in \left[\frac{w^a + w^b}{2}, w^b\right] \end{cases}.$$

Then, we have

1. $\max_{w \in [w^a, w^b]} u(w) - w^2 \leq \frac{1}{4} (w^b - w^a)^2$.
2. $\hat{u}(w) - w^2 \leq \frac{1}{2} (u(w) - w^2)$ for all $w \in [w^a, w^b]$.

Proof. See Appendix A.2. □

By replacing the square terms w_k^2 in (3.6) with the piecewise linear functions $u(w_k; W_k)$, we obtain an approximation problem

$$(3.10) \quad \begin{aligned} \bar{\vartheta}(W_1, \dots, W_k) &:= \min \quad \theta \\ \text{subject to} \quad &c_k^T x + \alpha_k + v_k^2 \leq u(w_k; W_k), \quad k \in [K], \\ &(x, \theta, \gamma, \pi, w, v) \in \mathcal{S}. \end{aligned}$$

PROPOSITION 3.3. Suppose that (3.6) has a finite optimum ϑ^* . Then, $\bar{\vartheta}(W_1, \dots, W_k) \leq \vartheta^*$.

Proof. See Appendix A.3. □

Next, we describe a mixed binary formulation to solve (3.10).

3.3. Mixed Binary Second Order Cone Program. A popular way to model a piecewise linear function is to use binary variables with SOS-2 constraints. To represent the piecewise linear function $u(w_k; W_k)$, we introduce binary variables $\mu_k \in \{0, 1\}^{N_k}$ and weight variables $\lambda_k \in [0, 1]^{N_k}$. Using μ_k and λ_k , we can write the piecewise linear function $u(w_k; W_k)$ as

$$(3.11) \quad u(w_k; W_k) = \sum_{j=1}^{N_k} \lambda_{kj} (w_k^j)^2$$

with a set of constraints

$$(3.12) \quad (w, \lambda_k, \mu_k) \in \mathcal{H}_k := \left\{ (w, \lambda_k, \mu_k) \left| \begin{array}{l} w = \sum_{j=1}^{N_k} \lambda_{kj} w_k^j, \sum_{j=1}^{N_k} \lambda_{kj} = 1, \sum_{j=1}^{N_k} \mu_{kj} \leq 2, \\ 0 \leq \lambda_{kj} \leq \mu_{kj}, j = 1, \dots, N_k, \mu_k \in \{0, 1\}^{N_k}, \\ \mu_{ki} + \mu_{kj} \leq 1, i, j = 1, \dots, N_k, i - j \geq 2 \end{array} \right. \right\},$$

where μ_{kj} and λ_{kj} represent the j^{th} elements of μ_k and λ_k , respectively. Using (3.11) and (3.12), we derive a mixed binary second order cone program (MB-SOCP) as

$$\begin{aligned}
 \bar{\vartheta}(W_1, \dots, W_k) &:= \min \quad \theta \\
 \text{subject to} \quad &c_k^T x + \alpha_k + v_k^2 \leq \sum_{j=1}^{N_k} \lambda_{kj} (w_k^j)^2, \quad k \in [K], \\
 &(w, \lambda_k, \mu_k) \in \mathcal{H}_k, \quad k \in [K], \\
 &(x, \theta, \gamma, \pi, w, v) \in \mathcal{S}.
 \end{aligned}
 \tag{3.13}$$

By exploiting the SOS-2 constraints (3.12), we are able to solve (3.13) in a branch and bound or a branch and cut framework. For example, we can develop a cutting plane algorithm that constructs disjunctive cuts using the variable transformation technique in (46). For detailed description of this cutting plane approach, see Appendix B. However, since the number of binary variables increases as our piecewise linear approximations get more accurate, it is computationally undesirable to solve (3.13) with a large number of evenly-spaced points in W_k . Alternatively, we can take an iterative approach which starts with a coarse piecewise linear approximation and then iteratively refines it using the optimal solution to (3.13). However, our computational experience reveals that solving a sequence of these mixed binary programs using an off-the-shelf solver such as Gurobi with limited control over the branching process can be computationally costly since for some branch and bound nodes the same computations are repeated across iterations.

4. Adaptive Branch and Bound Algorithm. In this section, we introduce a spatial branch and bound algorithm for solving (3.6). Rather than solving a sequence of mixed binary programs, our branch and bound algorithm adaptively divides the space of $(w_1, \dots, w_k)$ into smaller hyper-rectangles as it refines piecewise linear approximations. Specifically, starting with an initial hyper-rectangle $B_0 := [w_1^m, w_1^M] \times [w_2^m, w_2^M] \times \dots \times [w_K^m, w_K^M]$, the algorithm iteratively bisects a hyper-rectangle into two sub-hyper-rectangles and solves a second order cone program for each sub-hyper-rectangle. Using the optimal solution to the second order cone program, we update the lower bound and solve an evaluation problem to improve the incumbent solution. This branch and bound process is repeated until we obtain an ϵ -optimal solution, or meet some other termination criteria.

4.1. Initial Hyper-Rectangle. We first describe how we construct an initial hyper-rectangle B_0 . Using (3.3), we compute lower and upper bounds of w_k using bounds on γ_k and $z_k := d_k^T x + \beta_k$. To compute bounds on γ_k , we consider the Charnes-Cooper transformation (9). Using this variable transformation technique, we can compute a lower bound γ_k^m and an upper bound γ_k^M by solving the following LP problems:

$$(4.1) \quad \gamma_k^m = \min_{x \in \mathcal{X}} \frac{c_k^T x + \alpha_k}{d_k^T x + \beta} = \min_{y, t} c_k^T y + \alpha_k t \quad \text{subject to} \quad Ay \leq bt, \quad d_k^T y + \beta t = 1, \quad t \geq 0,$$

$$(4.2) \quad \gamma_k^M = \max_{x \in \mathcal{X}} \frac{c_k^T x + \alpha_k}{d_k^T x + \beta} = \max_{y, t} c_k^T y + \alpha_k t \quad \text{subject to} \quad Ay \leq bt, \quad d_k^T y + \beta t = 1, \quad t \geq 0.$$

On the other hand, we solve the following LP problems to obtain bounds on z_k :

$$(4.3) \quad z_k^m = \min_{x \in \mathcal{X}} d_k^T x + \beta_k, \quad z_k^M = \max_{x \in \mathcal{X}} d_k^T x + \beta_k.$$

Using the bounds on γ_k and z_k , we compute bounds on w_k as

$$(4.4) \quad w_k^m = \gamma_k^m + z_k^m, \quad w_k^M = \gamma_k^M + z_k^M.$$

319 **4.2. Second Order Cone Program.** Let $B := [w_1^a, w_1^b] \times [w_2^a, w_2^b] \times \cdots \times [w_K^a, w_K^b]$ be a
 320 sub-hyper-rectangle of B_0 such that $B \subset B_0$. For each $k \in [K]$, we consider a linear approximation
 321 of w_k^2 in $[w_k^a, w_k^b]$ as

$$322 \quad (4.5) \quad w_k^2 \leq (w_k^a + w_k^b)w_k - w_k^a w_k^b,$$

323 which results in the following second order cone constraint

$$324 \quad (4.6) \quad c_k^T x + \alpha_k + v_k^2 \leq (w_k^a + w_k^b)w_k - w_k^a w_k^b.$$

325 Having (4.6) in place of (3.7) and adding the box constraint $w \in B$ to (3.10), we obtain an SOCP
 326 approximation problem for the hyper-rectangle B as

$$327 \quad (4.7) \quad \begin{aligned} \bar{\vartheta}(B) := \min \quad & \theta \\ \text{subject to} \quad & c_k^T x + \alpha_k + v_k^2 \leq (w_k^a + w_k^b)w_k - w_k^a w_k^b, \quad k \in [K], \\ & w_k^a \leq w_k \leq w_k^b, \quad k \in [K], \\ & (x, \theta, \gamma, \pi, w, v) \in \mathcal{S}. \end{aligned}$$

328 **PROPOSITION 4.1.** Let $\vartheta^*(B)$ be the optimum of (3.6) with the additional box constraint $w \in B$.
 329 Then, we have $\bar{\vartheta}(B) \leq \vartheta^*(B)$.

330 *Proof.* This follows from (4.5). □

331 Since $\bar{\vartheta}(B)$ serves as a lower bound of $\vartheta^*(B)$, we can construct a lower bound of ϑ^* by taking
 332 the minimum of $\bar{\vartheta}(B)$ for all B in a partition of B_0 . This lower bound gets close to $\vartheta^*(B)$ as we
 333 refine the partition of B_0 .

334 **4.3. Evaluation Problem.** Let $(\bar{x}(B), \bar{\theta}(B), \bar{\gamma}(B), \bar{\pi}(B), \bar{w}(B), \bar{v}(B))$ be an optimal solution
 335 to (4.7) for a hyper-rectangle B . To evaluate the quality of $\bar{x}(B)$, we solve the following optimization
 336 problem:

$$337 \quad (4.8) \quad \begin{aligned} \psi(\bar{x}(B)) := \min \quad & \theta \\ \text{subject to} \quad & \frac{c_k^T \bar{x}(B) + \alpha_k}{d_k^T \bar{x}(B) + \beta_k} \leq \gamma_k, \quad k \in [K], \\ & f^T \pi \leq \theta, \quad H^T \pi \geq \gamma, \quad P\gamma \leq \theta \mathbb{1}_J, \\ & \theta \in \mathbb{R}, \quad \gamma \in \mathbb{R}^K, \quad \pi \in \mathbb{R}^L. \end{aligned}$$

338 Note that the evaluation of $\psi(\bar{x}(B))$ is equivalent to solving (1.1) with x fixed to $\bar{x}(B)$. Therefore,
 339 we have $\vartheta^* = \min_{x \in \mathcal{X}} \psi(x)$ and thus for any $x \in \mathcal{X}$, $\psi(x)$ serves as an upper bound of ϑ^* .

340 **4.4. Main Loop.** Starting with an initial hyper-rectangle B_0 , we solve (4.7) to obtain an
 341 optimal solution $\bar{x}(B_0)$ and its optimal objective value $\bar{\vartheta}(B_0)$. Then, we initialize t , x_{CB}^0 , ϑ_{CB}^0 , T_0
 342 as $t \leftarrow 0$, $x_{\text{CB}}^0 \leftarrow \bar{x}(B_0)$, $\vartheta_{\text{CB}}^0 \leftarrow \psi(\bar{x}(B_0))$, $T_0 \leftarrow \{B_0, \bar{x}(B_0), \bar{\vartheta}(B_0)\}$ where x_{CB}^t and ϑ_{CB}^t denote
 343 the incumbent solution at iteration t and its corresponding objective value, respectively.

344 At iteration t , we find an element $(B_t, \bar{x}(B_t), \bar{\vartheta}(B_t)) \in T_t$ such that $\bar{\vartheta}(B_t)$ is the smallest. Let

$$345 \quad B_t := [w_1^{a,t}, w_1^{b,t}] \times [w_2^{a,t}, w_2^{b,t}] \times \cdots \times [w_K^{a,t}, w_K^{b,t}], \quad \bar{x}^t \leftarrow \bar{x}(B_t), \quad \bar{\vartheta}^t \leftarrow \bar{\vartheta}(B_t).$$

346 If $\vartheta_{\text{CB}}^t - \bar{\vartheta}^t < \epsilon$, we terminate and return the incumbent solution x_{CB}^t and its objective value ϑ_{CB}^t .
 347 Otherwise, we find $k_t = \arg \max_k (w_k^{b,t} - w_k^{a,t})^2 / (4z_k^m)$ and divides B_t into

$$348 \quad (4.9) \quad B'_t := [w_1^{a,t}, w_1^{b,t}] \times \cdots \times [(w_{k_t}^{a,t} + w_{k_t}^{b,t})/2, w_{k_t}^{b,t}] \times \cdots \times [w_K^{a,t}, w_K^{b,t}],$$

$$349 \quad (4.10) \quad B''_t := [w_1^{a,t}, w_1^{b,t}] \times \cdots \times [w_{k_t}^{a,t}, (w_{k_t}^{a,t} + w_{k_t}^{b,t})/2] \times \cdots \times [w_K^{a,t}, w_K^{b,t}].$$

351 For B'_t and B''_t , we solve the associated SOCP relaxation problems (4.7) and obtain $(\bar{\vartheta}(B'_t), \bar{x}(B'_t))$
 352 and $(\bar{\vartheta}(B''_t), \bar{x}(B''_t))$. We update x_{CB}^{t+1} and $\vartheta_{\text{CB}}^{t+1}$ by comparing the current best objective value ϑ_{CB}^t
 353 with $\psi(\bar{x}(B'_t))$ and $\psi(\bar{x}(B''_t))$. We also update the branch and bound tree as

$$354 \quad T_{t+1} \leftarrow T_t \setminus \{(B_t, \bar{x}(B_t), \bar{\vartheta}(B_t))\} \cup \{(B'_t, \bar{x}(B'_t), \bar{\vartheta}(B'_t))\} \cup \{(B''_t, \bar{x}(B''_t), \bar{\vartheta}(B''_t))\}.$$

355 The above process is repeated until $\vartheta_{\text{CB}}^t - \bar{\vartheta}^t < \epsilon$ is satisfied or the iteration counter t reaches the iteration limit t_{max} . For a detail description, see Algorithm 4.1.

Algorithm 4.1 SOC-B

```

1: parameter: optimality tolerance  $\epsilon > 0$ , iteration limit  $t_{\text{max}}$ 
2: compute bounds on  $z_k$  and  $w_k$ 
3: construct an initial hyper-rectangle  $B_0$ 
4: solve (4.7) with  $B_0$  to obtain  $\bar{x}(B_0)$  and  $\bar{\vartheta}(B_0)$ 
5: let  $t \leftarrow 0$ ,  $x_{\text{CB}}^0 \leftarrow \bar{x}(B_0)$ ,  $\vartheta_{\text{CB}}^0 \leftarrow \psi(\bar{x}(B_0))$ ,  $T_0 \leftarrow \{(B_0, \bar{x}(B_0), \bar{\vartheta}(B_0))\}$ 
6: while  $t < t_{\text{max}}$  do
7:   find an element  $(B_t, \bar{x}(B_t), \bar{\vartheta}(B_t))$  in  $T_t$  such that  $\bar{\vartheta}(B_t)$  is the smallest
8:   let  $\bar{x}^t \leftarrow \bar{x}(B_t)$ ,  $\bar{\vartheta}^t \leftarrow \bar{\vartheta}(B_t)$ ,  $B_t := [w_1^{a,t}, w_1^{b,t}] \times [w_2^{a,t}, w_2^{b,t}] \times \cdots \times [w_K^{a,t}, w_K^{b,t}]$ 
9:   if  $\vartheta_{\text{CB}}^t - \bar{\vartheta}^t < \epsilon$  then
10:    return  $x_{\text{CB}}^t, \vartheta_{\text{CB}}^t$ 
11:   else
12:    let  $x_{\text{CB}}^{t+1} \leftarrow x_{\text{CB}}^t$ ,  $\vartheta_{\text{CB}}^{t+1} \leftarrow \vartheta_{\text{CB}}^t$ 
13:    find  $k_t = \arg \max_{k \in [K]} (w_k^{b,t} - w_k^{a,t})^2 / (4z_k^m)$ 
14:    construct  $B'_t, B''_t$  as (4.9), (4.10)
15:    for  $\bar{B}_t \in \{B'_t, B''_t\}$  do
16:      solve (4.7) with  $\bar{B}_t$  to obtain  $\bar{x}(\bar{B}_t), \bar{\vartheta}(\bar{B}_t)$ 
17:      if  $\psi(\bar{x}(\bar{B}_t)) < \vartheta_{\text{CB}}^{t+1}$  then
18:         $x_{\text{CB}}^{t+1} \leftarrow \bar{x}(\bar{B}_t)$ ,  $\vartheta_{\text{CB}}^{t+1} \leftarrow \psi(\bar{x}(\bar{B}_t))$ 
19:      end if
20:    end for
21:     $T_{t+1} \leftarrow T_t \setminus \{(B_t, \bar{x}(B_t), \bar{\vartheta}(B_t))\} \cup \{(B'_t, \bar{x}(B'_t), \bar{\vartheta}(B'_t))\} \cup \{(B''_t, \bar{x}(B''_t), \bar{\vartheta}(B''_t))\}$ 
22:    end if
23:     $t \leftarrow t + 1$ 
24: end while
    
```

356

357 **5. Convergence Analysis.** In this section, we provide a convergence analysis for SOC-B
 358 (Algorithm 4.1). Let M be the max of $\|P\|_\infty := \max_i \sum_j |P_{ij}|$ and $\max \{\|p\|_1 \mid p \in \mathcal{P}\}$. In order
 359 to show the finite convergence of SOC-B, we start with a theorem, which provides a bound on the
 360 gap between $\bar{\vartheta}^t$ and ϑ_{CB}^t as a function of the approximation errors at the optimal solution of (4.7)
 361 with $B = B_t$.

THEOREM 5.1. Let $(\bar{x}^t, \bar{\theta}^t, \bar{\gamma}^t, \bar{\pi}^t, \bar{w}^t, \bar{v}^t)$ and $\bar{\vartheta}^t$ be an optimal solution and the objective value to (4.7) with $B = B_t$. Then, we have

$$(5.1) \quad \vartheta_{CB}^t - \vartheta^* \leq \vartheta_{CB}^t - \bar{\vartheta}^t \leq M \max_{k \in [K]} \frac{\Delta_k}{z_k^m}.$$

where

$$(5.2) \quad \Delta_k := (w_k^{a,t} + w_k^{b,t})\bar{w}_k^t - w_k^{a,t}w_k^{b,t} - (\bar{w}_k^t)^2$$

represents an approximation error at $\bar{w}_k$ for $k \in [K]$.

Proof. From the feasibility of $(\bar{x}^t, \bar{\theta}^t, \bar{\gamma}^t, \bar{\pi}^t, \bar{w}^t, \bar{v}^t)$ to (4.7), we obtain

$$(5.3) \quad c_k^T \bar{x}^t + \alpha_k + (\bar{v}_k^t)^2 \leq (w_k^{a,t} + w_k^{b,t})\bar{w}_k^t - w_k^{a,t}w_k^{b,t}, \quad k \in [K].$$

Using (5.2), we further have

$$(5.3) \quad c_k^T \bar{x}^t + \alpha_k + (\bar{v}_k^t)^2 \leq (\bar{w}_k^t)^2 + \Delta_k, \quad k \in [K].$$

Since $\bar{\gamma}_k^t - (d_k^T \bar{x}^t + \beta_k) = 2\bar{v}_k^t$ and $\bar{\gamma}_k^t + (d_k^T \bar{x}^t + \beta_k) = 2\bar{w}_k^t$ for $k \in [K]$, we have

$$(5.4) \quad \bar{v}_k^t = \bar{w}_k^t - (d_k^T \bar{x}^t + \beta_k), \quad k \in [K].$$

Plugging (5.4) into (5.3), we have

$$(5.5) \quad c_k^T \bar{x}^t + \alpha_k + (d_k^T \bar{x}^t + \beta_k)^2 \leq 2\bar{w}_k^t(d_k^T \bar{x}^t + \beta_k) + \Delta_k, \quad k \in [K].$$

Consider a solution $(\check{\gamma}^t, \check{w}^t, \check{v}^t)$ such that

$$(5.6) \quad (\check{\gamma}_k^t, \check{w}_k^t, \check{v}_k^t) = \left(\bar{\gamma}_k^t + \frac{\Delta_k}{d_k^T \bar{x}^t + \beta_k}, \bar{w}_k^t + \frac{\Delta_k}{2(d_k^T \bar{x}^t + \beta_k)}, \bar{v}_k^t + \frac{\Delta_k}{2(d_k^T \bar{x}^t + \beta_k)} \right), \quad k \in [K].$$

From $\bar{\gamma}_k^t - (d_k^T \bar{x}^t + \beta_k) = 2\bar{v}_k^t$ and $\bar{\gamma}_k^t + (d_k^T \bar{x}^t + \beta_k) = 2\bar{w}_k^t$, we have $\check{\gamma}_k^t - (d_k^T \bar{x}^t + \beta_k) = 2\check{v}_k^t$ and $\check{\gamma}_k^t + (d_k^T \bar{x}^t + \beta_k) = 2\check{w}_k^t$. This implies $\check{v}_k^t = \check{w}_k^t - (d_k^T \bar{x}^t + \beta_k)$, which results in

$$(5.7) \quad (\check{w}_k^t)^2 - (c_k^T \bar{x}^t + \alpha_k + (\check{v}_k^t)^2) = 2\check{w}_k^t(d_k^T \bar{x}^t + \beta_k) - (c_k^T \bar{x}^t + \alpha_k + (d_k^T \bar{x}^t + \beta_k)^2).$$

Using (5.6) and (5.5), we have

$$(5.8) \quad 2\check{w}_k^t(d_k^T \bar{x}^t + \beta_k) - (c_k^T \bar{x}^t + \alpha_k + (d_k^T \bar{x}^t + \beta_k)^2) \geq 0.$$

From $\check{\gamma}_k^t - (d_k^T \bar{x}^t + \beta_k) = 2\check{v}_k^t$, $\check{\gamma}_k^t + (d_k^T \bar{x}^t + \beta_k) = 2\check{w}_k^t$, $(\check{w}_k^t)^2 - (c_k^T \bar{x}^t + \alpha_k + (\check{v}_k^t)^2) \geq 0$, we have

$$(5.7) \quad \frac{c_k^T \bar{x}^t + \alpha_k}{d_k^T \bar{x}^t + \beta_k} \leq \check{\gamma}_k^t = \bar{\gamma}_k^t + \frac{\Delta_k}{d_k^T \bar{x}^t + \beta_k}.$$

Let $(\hat{x}^t, \hat{\theta}^t, \hat{\gamma}^t, \hat{\pi}^t, \hat{w}^t, \hat{v}^t)$ be a solution such that

$$(5.8) \quad \hat{x}^t = \bar{x}^t, \quad \hat{\theta}^t = \bar{\theta}^t + \max_{k \in [K]} \frac{M\Delta_k}{d_k^T \bar{x}^t + \beta_k}, \quad \hat{\pi}^t \in \arg \min f^T \pi \text{ subject to } H^T \pi \geq \hat{\gamma}^t,$$

398 and

$$399 \quad (5.9) \quad (\hat{\gamma}_k^t, \hat{w}_k^t, \hat{v}_k^t) = \left(\frac{c_k^T \bar{x}^t + \alpha_k}{d_k^T \bar{x}^t + \beta_k}, \frac{\hat{\gamma}_k^t + (d_k^T \bar{x}^t + \beta_k)}{2}, \frac{\hat{\gamma}_k^t - (d_k^T \bar{x}^t + \beta_k)}{2} \right), \quad k \in [K].$$

401 Also, let $\hat{p}^t = \arg \max (\hat{\gamma}^t)^T p$ subject to $p \in \mathcal{P}$. Since $\mathcal{P}$ is assumed to be a polytope, the optimiza-
 402 tion problem, $\max (\hat{\gamma}^t)^T p$ subject to $p \in \mathcal{P}$, has a finite optimum, and thus $\hat{p}^t$ is well-defined. By
 403 (5.9), $(\hat{x}^t, \hat{\theta}^t, \hat{\gamma}^t, \hat{\pi}^t, \hat{w}^t, \hat{v}^t)$ satisfies (3.4) and $\gamma^m \leq \hat{\gamma}^t \leq \gamma^M$. Using (5.7) and (5.9), we have

$$404 \quad \hat{\gamma}_k^t \leq \bar{\gamma}_k^t + \frac{\Delta_k}{d_k^T \hat{x}^t + \beta_k}.$$

405 From the definition of M and the feasibility of $(\bar{x}^t, \bar{\theta}^t, \bar{\gamma}^t, \bar{\pi}^t, \bar{w}^t, \bar{v}^t)$ to (4.7), we obtain

$$406 \quad P\hat{\gamma}^t \leq P\bar{\gamma}^t + \max_{j \in [J]} \sum_{k=1}^K |P_{jk}| \cdot \max_{k \in [K]} \frac{\Delta_k}{d_k^T \bar{x}^t + \beta_k} \mathbf{1}_J \leq \bar{\theta}^t \mathbf{1}_J + \max_{k \in [K]} \frac{M\Delta_k}{d_k^T \bar{x}^t + \beta_k} \mathbf{1}_J = \hat{\theta}^t \mathbf{1}_J.$$

407 By the definition of $\hat{\pi}^t$, we have $H^T \hat{\pi}^t \geq \hat{\gamma}^t$. Moreover, we obtain

$$408 \quad f^T \hat{\pi}^t = (\hat{\gamma}^t)^T \hat{p}^t = (\bar{\gamma}^t)^T \hat{p}^t + \sum_{k=1}^K \frac{\hat{p}_k^t \Delta_k}{d_k^T \hat{x}^t + \beta_k} \leq f^T \bar{\pi}^t + \sum_{k=1}^K \frac{\hat{p}_k^t \Delta_k}{d_k^T \hat{x}^t + \beta_k} \leq \bar{\theta}^t + \max_{k \in [K]} \frac{M\Delta_k}{d_k^T \bar{x}^t + \beta_k} \leq \hat{\theta}^t.$$

409 The first equality holds from the strong duality and the first inequality follows due to weak duality
 410 since $H^T \bar{\pi}^t \geq \bar{\gamma}^t$ due to the feasibility of $(\bar{x}^t, \bar{\theta}^t, \bar{\gamma}^t, \bar{\pi}^t, \bar{w}^t, \bar{v}^t)$ to (4.7). So, $(\hat{x}^t, \hat{\theta}^t, \hat{\gamma}^t, \hat{\pi}^t, \hat{w}^t, \hat{v}^t)$ is
 411 feasible to (3.6) with the objective value of $\hat{\theta}^t$.

412 Since $(\hat{\theta}^t, \hat{\gamma}^t, \hat{\pi}^t)$ is feasible to (4.8) with respect to $\hat{x}^t$, we have $\vartheta^* \leq \psi(\hat{x}^t) \leq \hat{\theta}^t$. Since $w^* \in B_0$
 413 and no element of B_0 is excluded during the branching process in Algorithm 4.1, there exists some
 414 B_t^* such that $w^* \in B_t^*$. Since $(x^*, \theta^*, \gamma^*, \pi^*, w^*, v^*)$ is feasible to (4.7) with $B = B_t^*$, it follows that
 415 $\bar{\vartheta}(B_t^*) \leq \vartheta^*$. Also, by the selection rule of B_t and the way x_{CB}^t and ϑ_{CB}^t are updated, we have
 416 $\bar{\vartheta}(B_t) \leq \bar{\vartheta}(B_t^*)$ and $\vartheta_{CB}^t \leq \psi(\hat{x}^t)$. As a result, we have

$$417 \quad (5.10) \quad \vartheta_{CB}^t - \vartheta^* \leq \vartheta_{CB}^t - \bar{\vartheta}^t \leq \hat{\theta}^t - \bar{\theta}^t \leq \max_{k \in [K]} \frac{M\Delta_k}{d_k^T \bar{x}^t + \beta_k} \leq M \max_{k \in [K]} \frac{\Delta_k}{z_k^m}. \quad \square$$

419 Based on Theorem 5.1, we prove the finite convergence of SOC-B.

420 **THEOREM 5.2.** *For any $\epsilon > 0$, let*

$$421 \quad n = \sum_{k=1}^K n_k, \quad n_k = \left\lceil \log_2 \sqrt{\frac{M(w_k^M - w_k^m)^2}{4\epsilon z_k^m}} \right\rceil, \quad k \in [K].$$

422 *Algorithm 4.1 (SOC-B) terminates within 2^n iterations.*

423 *Proof.* By Theorem 5.1 and Proposition 3.2, we have

$$424 \quad \vartheta_{CB}^t - \bar{\vartheta}^t \leq M \max_{k \in [K]} \frac{\Delta_k}{z_k^m} \leq M \max_{k \in [K]} \frac{(w_k^{b,t} - w_k^{a,t})^2}{4z_k^m}.$$

425 Therefore, if $w_k^{b,t} - w_k^{a,t} \leq \sqrt{4\epsilon z_k^m / M}$ holds for all $k \in [K]$, the algorithm must terminate at iteration
 426 t by the termination criterion.

Let $\mathcal{B}$ be the set of 2^n hyper-rectangles obtained by dividing each $[w_k^m, w_k^M]$ into 2^{n_k} pieces of equal length of $(w_k^M - w_k^m)/2^{n_k}$ for every $k \in [K]$. Note that the hyper-rectangles in $\mathcal{B}$ are disjoint and their edges are not greater than $\sqrt{4\epsilon z_k^m/M}$. On the other hand, if the algorithm does not terminate through iteration t , by the termination criterion, we should have some $k' \in [K]$ such that $w_{k'}^{b,t} - w_{k'}^{a,t} > \sqrt{4\epsilon z_{k'}^m/M}$. Also, the hyper-rectangle B_t should not satisfy $w_k^{b,t} - w_k^{a,t} < (w_k^M - w_k^m)/2^{n_k}$ or $w_k^{b,t} - w_k^{a,t} \leq (w_k^M - w_k^m)/2^{n_k-1} \leq \sqrt{\epsilon z_k^m/M}$ for any $k \in [K]$. If there exists some k satisfying $w_k^{b,t} - w_k^{a,t} \leq \sqrt{\epsilon z_k^m/M}$, we have a contradiction to the branching rule since this implies

$$\frac{w_{k'}^{b,t} - w_{k'}^{a,t}}{\sqrt{z_{k'}^m}} > \sqrt{\frac{4\epsilon}{M}} \geq 2 \cdot \frac{w_k^{b,t} - w_k^{a,t}}{\sqrt{z_k^m}}.$$

Therefore, each hyper-rectangle B_t in Algorithm 4.1 should either be an element of $\mathcal{B}$ or a union of hyper-rectangles in $\mathcal{B}$ if the algorithm does not terminate through iteration t .

Suppose that the algorithm does not terminate before iteration 2^n . Since a hyper-rectangle is branched into two sub-hyper-rectangles at each iteration, at the start of iteration 2^n , we have $2^n + 1$ hyper-rectangles in T_{2^n} . However, since the hyper-rectangles in T_{2^n} are disjoint and they are either an element of $\mathcal{B}$ or a union of hyper-rectangles in $\mathcal{B}$, we have a contradiction to the fact that the number of hyper-rectangles in $\mathcal{B}$ is 2^n . Therefore, the algorithm should terminate within 2^n iterations. $\square$

Note that the number of iterations is in the order of $\mathcal{O}(\sqrt{\epsilon})$. This square root dependency provides an explanation for the improved performance of our SOCP-based approach over the LP-based benchmark algorithms.

COROLLARY 5.3. *For the following linear fractional programs:*

- *a stochastic linear fractional program with finite support or a sample average approximation to a stochastic linear fractional program (1.2) where $\sum_{k=1}^K p_k = 1$,*
- *a distributionally robust linear fractional program (6.2) with finite support and a polyhedral ambiguity set $\mathcal{P}$ where each element in $\mathcal{P}$ is a probability vector,*
- *a subproblem (6.12) of a distributionally robust linear fractional program with finite support where each row of P is a probability vector,*

we have $M = 1$. Therefore, Algorithm 4.1 (SOC-B) terminates within N iterations where

$$N = \sum_{k=1}^K \left\lceil \log_2 \sqrt{\frac{(w_k^M - w_k^m)^2}{4\epsilon z_k^m}} \right\rceil.$$

6. Distributionally Robust Linear Fractional Programming. In this section, we consider a distributionally robust linear fractional program with finite support $\xi^k = (c_k, d_k, \alpha_k, \beta_k)$:

$$(6.1) \quad \min_{x \in \mathcal{X}} \sup_{p \in \mathcal{U}} \sum_{k=1}^K p_k \frac{c_k^T x + \alpha_k}{d_k^T x + \beta_k}$$

where $p \in \mathbb{R}^K$ is a probability vector such that $\sum_{k=1}^K p_k = 1, p \geq 0$, and $\mathcal{U}$ is an ambiguity set of probability distributions. Assuming that (A1)-(A3) hold in (6.1), we introduce solution approaches for polyhedral and convex ambiguity sets below.

6.1. Polyhedral Ambiguity Set. Let $\mathcal{P}$ be a polyhedral ambiguity set (for the definition of $\mathcal{P}$, see (A5)). Using the LP duality, we derive the following reformulation.

PROPOSITION 6.1. *If the ambiguity set $\mathcal{U}$ is $\mathcal{P}$, the optimization problem (6.1) is equivalent to*

$$\begin{aligned} & \min \quad \theta \\ & \text{subject to} \quad \frac{c_k^T x + \alpha_k}{d_k^T x + \beta_k} \leq \gamma_k, \quad k \in [K], \\ & \quad f^T \pi \leq \theta, \quad H^T \pi \geq \gamma, \\ & \quad x \in \mathcal{X}, \quad \theta \in \mathbb{R}, \quad \gamma \in \mathbb{R}^K, \quad \pi \in \mathbb{R}^L. \end{aligned} \quad (6.2)$$

Proof. This follows directly from the linear programming duality.

THEOREM 6.2. *Algorithm 4.1 (SOC-B) finds an ϵ -optimal solution to (6.2) within N iterations.*

Proof. The optimization problem (6.2) is a special case of the linear fractional program (1.1). Since $\mathcal{P}$ is a polyhedral ambiguity set for probability vectors, every element p in $\mathcal{P}$ is a probability vector. By Corollary 5.3, we obtain the desired result. $\square$

Next, we present three formulations for ambiguity sets based on monomial moments, total variation distance, and Wasserstein metric. In what follows, let p^* denote the empirical distribution.

6.1.1. Moment-based Ambiguity Sets. Let $\mu(\xi) = (\mu^1(\xi), \mu^2(\xi), \dots, \mu^m(\xi))$ be a vector of moment functions on $\xi \in \mathbb{R}^{2(n+1)}$ where the i^{th} moment function μ^i is defined by some non-negative integers $d_1^i, d_2^i, \dots, d_{2(n+1)}^i$ such that $\mu^i(\xi) := (\xi_1)^{d_1^i} \cdot (\xi_2)^{d_2^i} \dots (\xi_{2(n+1)})^{d_{2(n+1)}^i}$. Given lower and upper bounds on μ and p , we can define a moment-based ambiguity set $\mathcal{P}_m$ as

$$\mathcal{P}_m := \left\{ p \mid \underline{\mu} \leq \sum_{k=1}^K p_k \mu(\xi^k) \leq \bar{\mu}, \quad \underline{p} \leq p \leq \bar{p} \right\}. \quad (6.3)$$

To ensure that p is a probability vector, we let $\underline{\mu}^1 = \bar{\mu}^1 = 1$ and $d_1^1 = d_2^1 = \dots = d_{2(n+1)}^1 = 0$.

PROPOSITION 6.3. *The optimization problem (6.1) with the moment-based ambiguity set $\mathcal{P}_m$ (6.3) is equivalent to*

$$\begin{aligned} & \min \quad -\underline{v}^T \underline{\mu} + \bar{v}^T \bar{\mu} - \underline{w}^T \underline{p} + \bar{w}^T \bar{p} \\ & \text{subject to} \quad (-\underline{v} + \bar{v})^T \mu(\xi^k) - \underline{w}_k + \bar{w}^k \geq \frac{c_k^T x + \alpha_k}{d_k^T x + \beta_k}, \quad k \in [K], \\ & \quad x \in \mathcal{X}, \quad \underline{v} \in \mathbb{R}_+^m, \quad \bar{v} \in \mathbb{R}_+^m, \quad \underline{w} \in \mathbb{R}_+^K, \quad \bar{w} \in \mathbb{R}_+^K. \end{aligned} \quad (6.4)$$

Proof. See (29, Theorem 1). $\square$

6.1.2. Total Variation Ambiguity Sets. Let p and q be probability measures over a finite set $\Omega := \{\xi^1, \xi^2, \dots, \xi^K\}$. We define the total variation distance between p and q as $d_{TV}(p, q) := \frac{1}{2} \sum_{k=1}^K |p_k - q_k|$ where $p_k := p(\xi^k)$ and $q_k := q(\xi^k)$ for $k \in [K]$. Given a radius $\Delta \geq 0$ and an empirical distribution $p^* \in \mathbb{R}_+^K$, we define a total variation ambiguity set $\mathcal{P}_{TV}$ as

$$\mathcal{P}_{TV} := \left\{ p \mid \frac{1}{2} \sum_{k=1}^K |p_k - p_k^*| \leq \Delta, \quad \sum_{k=1}^K p_k = 1, \quad p_k \geq 0, \quad k \in [K] \right\}. \quad (6.5)$$

PROPOSITION 6.4. The optimization problem (6.1) with the total variation ambiguity set $\mathcal{P}_{TV}$ (6.5) is equivalent to

$$\begin{aligned}
 & \min \quad \sum_{k=1}^K (s_k - t_k) p_k^* + \Delta \nu + \delta \\
 & \text{subject to} \quad s_k - t_k + \delta \geq \frac{c_k^T x + \alpha_k}{d_k^T x + \beta_k}, \quad \frac{1}{2} \nu - s_k - t_k \geq 0, \quad k \in [K], \\
 & \quad \quad \quad x \in \mathcal{X}, \quad s \in \mathbb{R}_+^K, \quad t \in \mathbb{R}_+^K, \quad \delta \in \mathbb{R}, \quad \nu \geq 0.
 \end{aligned}
 \tag{6.6}$$

Proof. This follows directly from the linear programming duality.

6.1.3. Wasserstein Ambiguity Sets. Let $d(\xi^i, \xi^j)$ be a distance between ξ^i and ξ^j . Given a radius $\Delta \geq 0$ and an empirical distribution $p^* \in \mathbb{R}_+^K$, the Wasserstein ambiguity set $\mathcal{P}_W$ is defined as

$$\mathcal{P}_W := \left\{ p \mid \exists q \in \mathbb{R}^{K \times K} : \begin{aligned} & \sum_{j=1}^K q_{ij} = p_i, \quad i \in [K], \quad \sum_{i=1}^K q_{ij} = p_j^*, \quad j \in [K], \\ & \sum_{k=1}^K p_k = 1, \quad p_k \geq 0, \quad k \in [K], \quad q_{ij} \geq 0, \quad i, j \in [K], \\ & \sum_{i=1}^K \sum_{j=1}^K q_{ij} d(\xi_i, \xi_j) \leq \Delta \end{aligned} \right\}.
 \tag{6.7}$$

PROPOSITION 6.5. The optimization problem (6.1) with the Wasserstein ambiguity set $\mathcal{P}_W$ (6.7) is equivalent to

$$\begin{aligned}
 & \min \quad - \sum_{k=1}^K p_k^* t_k - \Delta \nu + \delta \\
 & \text{subject to} \quad -s_k - r_k + \delta \geq \frac{c_k^T x + \alpha_k}{d_k^T x + \beta_k}, \quad k \in [K], \\
 & \quad \quad \quad -s_i + t_j + d(\xi_i, \xi_j) \nu + q_{ij} \leq 0, \quad i, j \in [K], \\
 & \quad \quad \quad x \in \mathcal{X}, \quad s \in \mathbb{R}^K, \quad t \in \mathbb{R}^K, \quad r \in \mathbb{R}_+^K, \quad \delta \in \mathbb{R}, \quad \nu \leq 0, \quad q \in \mathbb{R}_+^{K \times K}.
 \end{aligned}
 \tag{6.8}$$

Proof. See (29, Theorem 3). □

6.2. Convex Ambiguity Set. Since strong duality does not generally hold in convex optimization, for convex ambiguity set $\mathcal{C}$, we consider the following semi-infinite program

$$\begin{aligned}
 & \min_{x \in \mathcal{X}} \quad \theta \\
 & \text{subject to} \quad \frac{c_k^T x + \alpha_k}{d_k^T x + \beta_k} \leq \gamma_k, \quad k \in [K], \\
 & \quad \quad \quad \gamma^T p \leq \theta, \quad \forall p \in \mathcal{C}.
 \end{aligned}
 \tag{6.9}$$

Here are some examples of convex ambiguity sets.

6.2.1. Mean-Covariance Ambiguity Sets (10). Let $\mu \in \mathbb{R}^{2(n+1)}$ and $\Sigma \in \mathbb{R}^{2(n+1) \times 2(n+1)}$ be a reference mean vector and an invertible covariance matrix. Using the reference mean vector μ and covariance matrix Σ , we define an ambiguity set as

$$\mathcal{C}_{DY} := \left\{ p \mid \begin{aligned} & \left(\sum_{k=1}^K p_k \xi^k - \mu \right)^T \Sigma^{-1} \left(\sum_{k=1}^K p_k \xi^k - \mu \right) \leq \alpha, \\ & \sum_{k=1}^K p_k (\xi^k - \mu) (\xi^k - \mu)^T \preceq \beta \Sigma, \\ & \sum_{k=1}^K p_k = 1, \quad p_k \geq 0, \quad k \in [K]. \end{aligned} \right\}.
 \tag{6.10}$$

6.2.2. Divergence-based Ambiguity Sets. Let $\phi : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ be a convex divergence function. Popular choices of the function ϕ are $\phi(x) = x \ln(x) - x + 1$ (Kullback-Leibler), $\phi(x) = |x - 1|$ (total variation), $\phi(x) = (x - 1)^2$ (Modified χ^2 -distance) and $\phi(x) = (\sqrt{x} - 1)^2$ (Hellinger distance). For more ϕ -divergence functions, see (37, Table 1). Given the divergence function ϕ , a radius $\Delta \geq 0$, and an empirical distribution $p^* \in \mathbb{R}_+^K$, we define an ambiguity set $\mathcal{C}_\phi$ as

$$(6.11) \quad \mathcal{C}_\phi := \left\{ p \mid \sum_{k=1}^K \phi\left(\frac{p_k}{p_k^*}\right) p_k^* \leq \Delta, \sum_{k=1}^K p_k = 1, p_k \geq 0, k \in [K] \right\}.$$

Next, we discuss an approach to solve the semi-infinite program (6.9). To find an ϵ -optimal solution of (6.1), we present a cutting surface algorithm (32; 30), which iteratively solves

$$(6.12) \quad \begin{aligned} & \min_{x \in \mathcal{X}} \quad \theta \\ & \text{subject to} \quad \frac{c_k^T x + \alpha_k}{d_k^T x + \beta_k} \leq \gamma_k, \quad k \in [K], \\ & \quad \quad \quad P\gamma \leq \theta \mathbf{1}_t, \end{aligned}$$

where each row of P is an element of a finite set $\mathcal{C}^t := \{p^*, p^0, \dots, p^{t-1}\} \subset \mathcal{C}$. Each time we obtain an ϵ -optimal solution $(x^t, \gamma^t, \theta^t)$ to (6.12), we solve the separation problem $\max_{p \in \mathcal{C}} p^T \gamma^t$ to either terminate or generate a probability cut p^t . For details, see Algorithm 6.1.

Algorithm 6.1 A cutting surface algorithm for (6.1)

- 1: Input: optimality tolerance $\epsilon > 0$, empirical distribution p^* .
 - 2: Step 1: $\mathcal{C}^0 \leftarrow \{p^*\}$, $t \leftarrow 0$.
 - 3: Step 2: Determine an optimal solution $(x^t, \gamma^t, \theta^t)$ of (6.12) with $\mathcal{C}^t$.
 - 4: Step 3: Determine an $\epsilon/2$ -optimal solution p^t of the problem $\max_{p \in \mathcal{C}} p^T \gamma^t$.
 - 5: Step 4: If $(p^t)^T \gamma^t - \theta^t \leq \epsilon/2$, stop and return x^t and θ^t ; otherwise $\mathcal{C}^{t+1} \leftarrow \mathcal{C}^t \cup \{p^t\}$, $t \leftarrow t + 1$, and go to Step 2.
-

Let $\theta^M := \max_{k \in [K]} \gamma_k^M$ and $\Gamma := \{(x, \gamma, \theta) \mid x \in \mathcal{X}, (c_k^T x + \alpha_k)/(d_k^T x + \beta_k) \leq \gamma_k \leq \gamma_k^M, k \in [K], 0 \leq \theta \leq \theta^M\}$ (for the definition of γ_k^M , see (4.2)).

PROPOSITION 6.6. *Let $(x^t, \gamma^t, \theta^t)$ be an optimal solution obtained by solving (6.12) with $\mathcal{C}^t$ as in Algorithm 6.1. For any $t \geq 0$, $(x^t, \gamma^t, \theta^t)$ belongs to the compact set Γ .*

Proof. Since matrix P constructed from $\mathcal{C}^t$ is non-negative, Assumption (A4) is satisfied. Therefore, using Proposition 3.1, let $(x^t, \gamma^t, \theta^t)$ satisfy $\gamma_k^t = (c_k^T x^t + \alpha_k)/(d_k^T x^t + \beta_k)$ for all $k \in [K]$. By the definition of γ_k^M in (4.2), we have $\gamma_k^t \leq \gamma_k^M$ for $k \in [K]$. Also, by (A2)-(A3), we have $(c_k^T x + \alpha_k)/(d_k^T x + \beta_k) \geq 0$ for all $x \in \mathcal{X}$, leading to $\gamma_k^t \geq 0$ for $k \in [K]$. Therefore, $0 \leq \theta^t$ since $P \geq 0$.

Next, suppose that $\theta^t > \theta^M$. If $p^T \gamma^t \leq \theta^M$ for all $p \in \mathcal{C}^t$, we have a contradiction since we can improve the objective value by decreasing θ^t to θ^M . Therefore, there should exist some $\hat{p} \in \mathcal{C}^t$ such that $\hat{p}^T \gamma^t > \theta^M$. Since $\hat{p}$ is a probability measure, it implies that $\theta^M = \max_{k \in [K]} \gamma_k^M \geq \max_{k \in [K]} \gamma_k^t \geq \hat{p}^T \gamma^t > \theta^M$, which leads to a contradiction. Therefore, we have $\theta^t \leq \theta^M$, resulting in $(x^t, \gamma^t, \theta^t) \in \Gamma$. The set Γ is compact since it is closed and bounded. $\square$

THEOREM 6.7. *Let $\mathcal{C}$ be a compact set. Algorithm 6.1 returns an ϵ -optimal solution in a finite number of iterations.*

Proof. Since Γ and $\mathcal{C}$ are compact, so is $\Gamma \times \mathcal{C}$. Since $g(x, \gamma, \theta, p) := \gamma^T p - \theta$ is continuous on $\Gamma \times \mathcal{C}$, by Proposition 6.6 and the argument in (28, Theorem 4.2), we obtain the desired result. $\square$

7. Numerical Experiments. In this section, we report numerical experiments of the proposed algorithms for the linear fractional program (1.2) and the distributionally robust linear fractional program with finite support (6.1).

7.1. Linear Fractional Program. We compare the proposed algorithm (SOC-B) with benchmark algorithms on diverse examples to test the scalability with respect to the number of variables n , the number of fractional terms K , and the optimality tolerance ϵ . For the benchmark algorithms, we select two branch and bound algorithms (26; 19) and a polyblock-based outer approximation algorithm (35). For all branch and bound algorithms, we adopt the rule that branches a node having the best bound. By examining how relative optimality gaps change over time, we compare the performance of the algorithms.

Two types of test examples are considered in the experiments. First, we considered small examples that have been introduced in the literature (13; 26; 35; 34; 19). Second, we generated synthetic examples by varying the values of n and K to test the scalability of the algorithms. For the experiments, we used a 64-core server with Xeon 2.20 GHz CPUs and 128 GB RAM where each core is used to run an algorithm on a test example. All the codes are written in Python and use Gurobi to solve base problems (LP, SOCP).

7.1.1. Small Examples from the Literature. We consider six examples from the literature. These examples are from Falk (13), Pei I,II (34), Phuong (35), and Kuno I,II (26) with the size (n, K) of (2, 2), (3, 3), (3, 4), (12, 5), (3, 3), and (3, 4), respectively.

Experience with Commercial Solvers. We attempted to solve (3.6) using two commercially available solvers Gurobi and GloMIQO (33), which are developed to solve bilinear optimization models. GloMIQO was used from the GAMS interface. Gurobi failed to find a meaningful solution for all cases. On the other hand, GloMIQO (33) found an optimal solution for all cases using its local solver. However, the reported gap was more than 100% even after running the solver for 24 hours. This computational experience suggests the need for specialized algorithms to solve linear fractional programs.

Table 1: Experimental results of Gurobi and GloMIQO (33) on six small examples in the literature. After running the algorithms for 24 hours, we report computation times (in seconds), best feasible objective values, and best bounds.

Problem	Gurobi			GloMIQO		
	Run Time	Best Feasible	Best Bound	Run Time	Best Feasible	Best Bound
Falk	86436.13	-	-	86401.11	-5.0000	-1.90e+06
Pei I	86404.65	1.0247e+10	-	86401.08	-3.0000	-3.83e+06
Pei II	86663.79	-	-	86401.08	-4.0907	-5.75e+06
Phuong	86410.24	-	-	86417.00	-16.0576	-9.99e+06
Kuno I	86409.19	1.3318e+10	-	86400.93	-3.0029	-3.82e+06
Kuno II	86527.22	-	-	86401.32	3.7984	-5.74e+06

Experimental Results with Specialized Algorithms. We now report computational experiments with our implementation of the proposed algorithm and other special algorithms developed in the literature. We implemented Kuno’s algorithm (26), Jiao and Liu’s algorithm (19) and Phuong

and Tuy's algorithm (35). For each problem, we run the algorithms until they reach a (relative) optimality gap of 10^{-5} and report computation times taken to attain an ϵ -optimality gap for $\epsilon \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. Table 2 gives the experimental results. The results in Table 2 show that SOC-B and Kuno's algorithm (26) are the fastest algorithms for these examples. Both of them take less than three seconds to reach an optimality gap of 10^{-5} for all examples. While Kuno's algorithm is faster than SOC-B when $n \leq 3$, the performance gap decreases as n increases and ϵ gets small as seen in the Phuong instance. The Phuong and Tuy's algorithm (35) is competitive to Kuno's algorithm and SOC-B for the Falk, Pei I, Pei II, and Kuno II instances. However, it is slower than Kuno's algorithm and SOC-B for the Kuno I instance. For the Phuong instance, the Phuong and Tuy's algorithm's solution time is significantly greater, and it performs even worse than the Jiao and Liu's algorithm (19). Except for the Phuong instance, the Jiao and Liu's algorithm is the slowest algorithm.

Table 2: Experimental results for the four specialized algorithms on six small examples in the literature. Computation times (in seconds) are for attaining a relative optimality gap tolerance in the range 10^{-2} to 10^{-5} .

Algorithm	Tolerance	Falk	Pei I	Pei II	Phuong	Kuno I	Kuno II
SOC-B (Algorithm 4.1)	10^{-2}	0.31	0.62	0.26	1.37	0.51	0.32
	10^{-3}	0.47	0.92	0.33	1.64	1.39	0.58
	10^{-4}	0.58	1.01	0.50	1.89	2.27	0.84
	10^{-5}	0.62	1.08	0.63	2.02	2.65	0.97
Kuno (26)	10^{-2}	0.03	0.01	0.03	0.33	0.02	0.03
	10^{-3}	0.05	0.01	0.03	0.63	0.10	0.03
	10^{-4}	0.05	0.04	0.08	1.40	0.19	0.09
	10^{-5}	0.05	0.08	0.08	1.89	0.22	0.09
Jiao and Liu (19)	10^{-2}	1.33	2.09	0.90	8.15	3.10	2.33
	10^{-3}	4.25	11.00	3.34	24.66	30.37	11.42
	10^{-4}	7.96	23.64	6.41	54.18	94.22	23.22
	10^{-5}	11.89	36.08	9.19	90.60	164.82	34.57
Phuong and Tuy (35)	10^{-2}	0.16	0.05	0.04	642.64	0.21	0.02
	10^{-3}	0.43	0.22	0.08	3520.37	2.38	0.09
	10^{-4}	0.66	0.36	0.15	6208.80	7.53	0.15
	10^{-5}	0.86	0.50	0.22	15319.08	13.11	0.23

Although these small examples are widely used for comparing the algorithms in the literature, they have the following limitations. First, α_k and β_k are relatively large in comparison with c_k and d_k . This makes linear fractional functions have little curvature. Second, the problem size (n, K) is small. Since the number of auxiliary variables is increasing with K in all algorithms, problems with large K should be experimented to test the scalability of the algorithms. In order to better evaluate the algorithms, we generate synthetic problem instances as follows.

7.1.2. Synthetic Problem Instances. Let a_{ij} be the entry in the i^{th} row and j^{th} column of a matrix A . We generate the problem parameters $c_k, d_k, a_i \in \mathbb{R}^n$ and $\alpha_k, \beta_k, p_k, b_i \in \mathbb{R}$ where $i \in [m]$ and $k \in [K]$ in (1.2) as $n, K \in \{5, 10, 25, 50, 100\}$, $m = \lceil n/2 \rceil$, $p_k = 1/K$, $a_{ij} \sim \text{Uniform}(1, n)$, $b_i = n$, $c_{kj} \sim \text{Uniform}(-1, 1)$, $\bar{\alpha}_k \sim \text{Uniform}(-1, 0)$, $\alpha_k = \bar{\alpha}_k - \max_{x \in \mathcal{X}} c_k^T x$, $d_{kj} \sim \text{Uniform}(-1, 1)$, $\bar{\beta}_k \sim \text{Uniform}(0, 1)$, and $\beta_k = 1 - \min_{x \in \mathcal{X}} d_k^T x$. From $x \geq 0$, $a_{ij} \sim \text{Uniform}(1, n)$, and $b_i = n$, we have $0 \leq x_j \leq 1$. It ensures that $\mathcal{X}$ is a polytope. From the definitions of α_k and β_k , we have

593 $c_k^T x + \alpha_k \leq \bar{\alpha}_k < 0$ and $d_k^T x + \beta_k \geq \bar{\beta}_k > 0$. Our data generation model allows both signs for c_{kj}
 594 and d_{kj} , contrary to the data generation models in (19; 26) which allows only positive values for
 595 c_{kj} and d_{kj} . Allowing both signs for c_{kj} and d_{kj} , we can generate more difficult linear fractional
 596 functions.

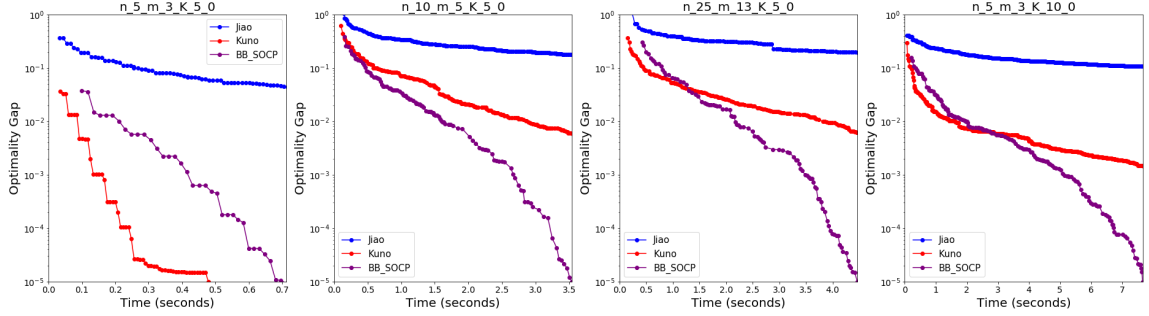


Fig. 2: Objective value convergence plots for a problem instance of the size $(n, K) \in \{(5, 5), (10, 5), (25, 5), (5, 10)\}$.

597 Table 3 gives the results for synthetic instances. For each choice of (n, K) , we generate five
 598 instances and run the algorithms until they reach an optimality gap of 10^{-5} with the time limit
 599 of 24 hours. If a certain level of optimality gap is reached for all five instances, we report the
 600 average of computation times. The Phuong and Tuy's algorithm takes a large time as the size
 601 of n and K increases. So, the experimental results of this algorithm are not included in the
 602 table. Results in Table 3 show that SOC-B scales much better as the size of n and K increases.
 603 When n and K are small, SOC-B and Kuno's algorithm take a similar amount of time to reach
 604 an optimality gap of 10^{-2} as shown in the instances with $(n, K) \in \{(5, 5), (10, 5), (25, 5), (5, 10)\}$.
 605 However, although computation times to reach an optimality gap of 10^{-2} are similar between
 606 SOC-B and Kuno's algorithm for $(n, K) \in \{(10, 5), (25, 5), (5, 10)\}$, the optimality gap of SOC-B
 607 decreases more quickly, as illustrated in Figure 2. This is attributed to the square root dependency
 608 on ϵ of SOC-B, which is not seen in LP-based approximations. For the instances with $(n, K) \notin$
 609 $\{(5, 5), (10, 5), (25, 5), (5, 10)\}$, SOC-B clearly outperforms the benchmark algorithms. For these
 610 instances SOC-B takes much smaller computation times to attain an optimality gap of 10^{-2} . The
 611 efficiency of SOC-B over the benchmark algorithms sharply increases as ϵ gets small. To attain an
 612 optimality gap of 10^{-5} , SOC-B is at least five times faster than the benchmark algorithms. Contrary
 613 to the other algorithms, SOC-B can solve all five instances of $(n, K) = (10, 25)$ and $(100, 10)$ within
 614 the time limit. For $(n, K) = (25, 25)$ and $(5, 50)$, SOC-B achieves an optimality gap of 10^{-5} for two
 615 out of the five instances, as shown in Table 4.

616 In summary, SOC-B attains two or more digit accuracy faster than the benchmark algorithms
 617 in most cases. In particular, it efficiently finds a highly accurate solution due to the square root
 618 dependency on ϵ .

619 **7.2. Distributionally Robust Linear Fractional Program.** We present experimental res-
 620 ults for the distributionally robust linear fractional program with finite support (6.1). For each
 621 $(n, K) \in \{(5, 10), (10, 10), (25, 10), (50, 10), (100, 10), (5, 25), (10, 25)\}$, we randomly choose one
 622 instance among the five instances used in Section 7.1.2. We use the Wasserstein ambiguity set (6.7)

Table 3: Experimental results for three specialized algorithms on synthetic problem instances. For each (n, K) , we run the algorithms on five instances with a 24-hour time limit. We report average computation times (in seconds) over five instances to reach a range of relative optimality gap tolerances from 10^{-2} to 10^{-5} . A blank cell represents that an algorithm could not attain a target relative optimality gap tolerance within the time limit in any of the five instances.

Problem		SOC-B (Algorithm 4.1)				Kuno (26)				Jiao and Liu (19)			
K	n	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
5	5	0.76	1.19	1.47	1.79	0.35	0.65	0.87	1.16	6.40	19.95	32.60	48.41
	10	1.38	2.30	2.69	3.07	1.62	3.49	5.37	7.01	72.55	221.26	406.19	594.15
	25	3.16	4.30	4.88	5.37	4.87	12.14	23.18	36.64	122.10	423.97	961.19	1647.64
	50	11.58	19.00	22.48	23.88	33.27	178.52	604.00	1216.33	5052.41	-	-	-
	100	53.27	77.28	86.79	91.42	203.93	1117.85	5669.88	20264.29	21779.47	-	-	-
10	5	7.93	13.80	16.22	17.44	8.79	20.54	28.62	34.66	32152.78	-	-	-
	10	12.92	21.70	24.84	25.76	25.53	68.52	102.36	130.53	-	-	-	-
	25	56.91	81.61	88.20	90.43	242.38	690.48	1125.68	1511.42	-	-	-	-
	50	211.90	322.90	352.59	362.73	1208.87	3859.17	12409.12	-	-	-	-	-
	100	1290.59	2087.42	2294.36	2333.53	22261.21	-	-	-	-	-	-	-
25	5	190.61	419.30	477.57	488.98	316.76	1122.55	1773.37	2377.53	-	-	-	-
	10	4828.92	8260.64	8748.70	8811.11	-	-	-	-	-	-	-	-

Table 4: Experimental results of three specialized algorithms for synthetic instances with large K . For each $(n, K) = (25, 25), (5, 50)$, we run the algorithms on five instances until a relative optimality tolerance of 10^{-5} is satisfied with a 24-hour time limit. We report computation times (in seconds) and relative optimality gaps for all instances.

Problem			SOC-B (Algorithm 4.1)		Kuno (26)		Jiao and Liu (19)	
K	n	#	Time (s)	Opt. Gap	Time (s)	Opt. Gap	Time (s)	Opt. Gap
25	25	1	86400	0.038	86400	0.064	86400	0.22
		2	65735	1.00e-05	86400	0.042	86400	0.21
		3	74886	1.00e-05	86400	0.047	86400	0.17
		4	86400	0.067	86400	0.070	86400	0.24
		5	86400	0.011	86400	0.053	86400	0.19
50	5	1	86400	0.029	86400	0.044	86400	0.28
		2	40359	1.00e-05	86400	0.020	86400	0.22
		3	6334	1.00e-05	86400	0.0015	86400	0.21
		4	86400	0.018	86400	0.038	86400	0.25
		5	86400	0.030	86400	0.045	86400	0.36

with $d(\xi_i, \xi_j) = \|\xi_i - \xi_j\|_1$. For a radius $\Delta \geq 0$, we set $\Delta = \rho \Delta^{\max}$ where $\rho \in \{0, 0.01, 0.05, 0.1, 0.2\}$ and $\Delta^{\max}$ is the minimum distance such that all probability distributions supported on the finite support are included in $\mathcal{P}^W$ when $\Delta = \Delta^{\max}$. We consider the two proposed solution approaches (dual reformulation, cutting surface algorithm) to solve the optimization problem. Table 5 gives the results for distributionally robust linear fractional programs.

Table 5 suggests that the two approaches solve most instances within the time limit of 24 hours. The dual reformulation approach is more efficient than the cutting surface algorithm since it solves only one instance of (1.1). As the radius Δ increases, computational times tend to increase in both approaches. The number of cuts also grow with Δ . In the cutting surface algorithm, while two or fewer probability cuts are generated for $\rho \leq 0.05$, this number increases to more than 5 when $\rho \geq 0.1$. On the other hand, objective values drop more quickly for large K . For $\rho = 0.01$

Table 5: Experimental results of two solution approaches for distributionally robust linear fractional programs. For each (n, K) , we run the algorithms with varying ρ from 0 to 0.2 until a relative optimality gap of 10^{-5} is obtained with a 24-hour time limit. We report objective values, computation times, relative optimality gaps and the number of probability cuts (for cutting surface algorithm only).

Problem			Dual			Cutting Surface			
K	n	ρ	Obj. Val	Time (s)	Opt. Gap	Obj. Val	Time (s)	Opt. Gap	Cuts
10	5	0	-1.1493	37.56	1.00E-05	-1.1493	46.76	1.00E-05	0
		0.01	-1.1302	42.22	1.00E-05	-1.1302	60.75	1.00E-05	1
		0.05	-1.0536	25.05	1.00E-05	-1.0536	37.33	1.00E-05	1
		0.1	-0.9591	59.23	1.00E-05	-0.9591	123.74	1.00E-05	4
		0.2	-0.8479	495.6	1.00E-05	-0.8479	3144.24	1.00E-05	12
	10	0	-1.3251	44.37	1.00E-05	-1.3251	55.62	1.00E-05	0
		0.01	-1.3003	31.84	1.00E-05	-1.3003	45.7	1.00E-05	1
		0.05	-1.2009	41.59	1.00E-05	-1.2009	60.57	1.00E-05	1
		0.1	-1.0929	129.33	1.00E-05	-1.0929	266.57	1.00E-05	2
		0.2	-0.9556	351.06	1.00E-05	-0.9556	9502.88	1.00E-05	9
	25	0	-0.9717	290.59	1.00E-05	-0.9717	378.06	1.00E-05	0
		0.01	-0.9458	921.47	1.00E-05	-0.9458	1763.99	1.00E-05	2
		0.05	-0.8934	1182.96	1.00E-05	-0.8934	4363.21	1.00E-05	6
		0.1	-0.8483	1359.33	1.00E-05	-0.8483	7541.36	1.00E-05	9
		0.2	-0.7906	2190.53	1.00E-05	-0.7906	30447.53	1.00E-05	25
	50	0	-1.0661	1076.99	1.00E-05	-1.0661	1742.33	1.00E-05	0
		0.01	-1.0518	1988.09	1.00E-05	-1.0518	4543.3	1.00E-05	2
		0.05	-0.9976	3399.73	1.00E-05	-0.9976	10338.17	1.00E-05	4
		0.1	-0.9607	3755.13	1.00E-05	-0.9607	22570.45	1.00E-05	9
		0.2	-0.9302	2264.33	1.00E-05	-0.9302	25342.39	1.00E-05	16
	100	0	-1.1906	2421.99	1.00E-05	-1.1906	4423.74	1.00E-05	0
		0.01	-1.179	3337.54	1.00E-05	-1.179	8594.49	1.00E-05	1
		0.05	-1.138	2909.73	1.00E-05	-1.138	9598.92	1.00E-05	1
		0.1	-1.0908	3679.35	1.00E-05	-1.0908	26938.46	1.00E-05	6
		0.2	-1.0455	2901.88	1.00E-05	-1.0455	58606.53	1.00E-05	13
25	5	0	-1.3338	5929.86	1.00E-05	-1.3338	6621.21	1.00E-05	0
		0.01	-1.2729	28455.42	1.00E-05	-1.2729	30113.09	1.00E-05	1
		0.05	-1.0825	59026.49	1.00E-05	-0.9676	66166.13	0.1675	-
		0.1	-0.8974	64801.5	3.54E-02	-0.8942	65462.24	0.4396	-
		0.2	-0.6791	64800.95	9.03E-02	-0.6405	65214.21	0.6933	-
	10	0	-1.2096	2313.82	1.00E-05	-1.2096	2625.93	1.00E-05	0
		0.01	-1.1579	12977.19	1.00E-05	-1.1579	13824.77	1.00E-05	1
		0.05	-0.9648	36695.22	1.00E-05	-0.9648	48759.61	1.00E-05	2
		0.1	-0.7704	64801.91	9.53E-02	-0.7606	65179.75	1.23E-01	-
		0.2	-0.637	64802.66	0.4490	-0.4945	65106.69	0.7151	-

and 0.05, the objective values decrease by 3% and 10% when $K = 10$. However, for $K = 25$, the objective values decrease by 5% and 20%, respectively. The dual reformulation approach solves small instances with K up to 25 within a reasonable amount of time.

8. Concluding Remarks. In this paper we consider a linear fractional model in a form which allows us to consider the linear fractional program and its stochastic and distributionally robust counterparts in the same framework. Based on second order cone approximations, we develop an adaptive branch and bound algorithms to solve this model. Our branch and bound algorithm has an attractive property that the number of iterations to attain an ϵ -optimal solution is in the order

of $\mathcal{O}(\sqrt{\epsilon})$. Due to this property, the algorithm finds an accurate solution faster than the LP-based benchmark algorithms. The algorithm scales better with problem size and the number of terms in the fractional program. Specifically, it can solve small size instances with K up to 50 and medium size instances with K up to 25. This result is important when fractional programs are obtained from the sample average approximations of a stochastic program. For distributionally robust linear fractional programs, our reformulation approach can solve small size instances with the cardinality of support up to 25.

Acknowledgments. We would like to thank the associate editor and two anonymous referees who kindly reviewed the earlier versions of this manuscript and provided valuable suggestions and comments.

References.

- [1] Y. ALMOGY AND O. LEVIN, *Parametric Analysis of a Multi-Stage Stochastic Shipping Problem*, Operational Research, 69 (1970), pp. 359–370.
- [2] Y. ALMOGY AND O. LEVIN, *A Class of Fractional Programming Problems*, Operations Research, 19 (1971), pp. 57–67.
- [3] A. BEN-TAL, D. DEN HERTOOG, A. DE WAEGENAERE, B. MELENBERG, AND G. RENNEN, *Robust Solutions of Optimization Problems Affected by Uncertain Probabilities*, Management Science, 59 (2013), pp. 341–357.
- [4] H. P. BENSON, *On the Global Optimization of Sums of Linear Fractional Functions over a Convex Set*, Journal of Optimization Theory and Applications, 121 (2004), pp. 19–39.
- [5] D. BERTSIMAS, X. V. DOAN, K. NATARAJAN, AND C.-P. TEO, *Models for Minimax Stochastic Linear Optimization Problems with Risk Aversion*, Mathematics of Operations Research, 35 (2010), pp. 580–602.
- [6] J. R. BIRGE AND R. J.-B. WETS, *Computing Bounds for Stochastic Programming Problems by Means of a Generalized Moment Problem*, Mathematics of Operations Research, 12 (1987), pp. 149–162.
- [7] G. C. CALAFIORE, *Ambiguous Risk Measures and Optimal Robust Portfolios*, SIAM Journal on Optimization, 18 (2007), pp. 853–877, <https://doi.org/10.1137/060654803>.
- [8] A. CAMBINI, L. MARTEIN, AND S. SCHAIBLE, *On Maximizing a Sum of Ratios*, Journal of Information and Optimization Sciences, 10 (1989), pp. 65–79.
- [9] A. CHARNES AND W. W. COOPER, *Programming with Linear Fractional Functionals*, Naval Research Logistics Quarterly, 9 (1962), pp. 181–186.
- [10] E. DELAGE AND Y. YE, *Distributionally Robust Optimization under Moment Uncertainty with Application to Data-Driven Problems*, Operations Research, 58 (2010), pp. 595–612.
- [11] W. DINKELBACH, *On Nonlinear Fractional Programming*, Management Science, 13 (1967), pp. 492–498.
- [12] P. M. ESFAHANI AND D. KUHN, *Data-Driven Distributionally Robust Optimization Using the Wasserstein Metric: Performance Guarantees and Tractable Reformulations*, Mathematical Programming, 171 (2018), pp. 115–166.
- [13] J. E. FALK AND S. W. PALOCSAY, *Image Space Analysis of Generalized Fractional Programs*, Journal of Global Optimization, 4 (1994), pp. 63–88.
- [14] R. GAO AND A. J. KLEYWEGT, *Distributionally Robust Stochastic Optimization with Wasserstein Distance*, arXiv preprint arXiv:1604.02199, (2016).
- [15] N. GOGIA, *The Multiplex Method for Linear Fractional Programming*, Cahiers du Centre d’Etudes de Recherche Operationnelle, 9 (1967), pp. 123–133.

- 687 [16] P. HARTMAN ET AL., *On Functions Representable as a Difference of Convex Functions.*, Pacific
688 Journal of Mathematics, 9 (1959), pp. 707–713.
- 689 [17] T. HOMEM-DE MELLO AND S. MEHROTRA, *A Cutting-Surface Method for Uncertain Linear*
690 *Programs with Polyhedral Stochastic Dominance Constraints*, SIAM Journal on Optimization,
691 20 (2009), pp. 1250–1273.
- 692 [18] R. JIANG AND Y. GUAN, *Risk-Averse Two-Stage Stochastic Program with Distributional Am-*
693 *biguity*, Operations Research, 66 (2018), pp. 1390–1405.
- 694 [19] H.-W. JIAO AND S.-Y. LIU, *A Practicable Branch and Bound Algorithm for Sum of Linear*
695 *Ratios Problem*, European Journal of Operational Research, 243 (2015), pp. 723–730.
- 696 [20] H. KONNO AND N. ABE, *Minimization of the Sum of Three Linear Fractional Functions*,
697 Journal of Global Optimization, 15 (1999), pp. 419–432.
- 698 [21] H. KONNO AND K. FUKAISHI, *A Branch and Bound Algorithm for Solving Low Rank Lin-*
699 *ear Multiplicative and Fractional Programming Problems*, Journal of Global Optimization, 18
700 (2000), pp. 283–299.
- 701 [22] H. KONNO AND H. WATANABE, *Bond Portfolio Optimization Problems and Their Applications*
702 *to Index Tracking: A Partial Optimization Approach*, Journal of the Operations Research
703 Society of Japan, 39 (1996), pp. 295–306.
- 704 [23] H. KONNO, Y. YAJIMA, AND T. MATSUI, *Parametric Simplex Algorithms for Solving a Special*
705 *Class of Nonconvex Minimization Problems*, Journal of Global Optimization, 1 (1991), pp. 65–
706 81.
- 707 [24] H. KONNO AND H. YAMASHITA, *Minimizing Sums and Products of Linear Fractional Functions*
708 *over a Polytope*, Naval Research Logistics, 46 (1999), pp. 583–596.
- 709 [25] S. KRUK AND H. WOLKOWICZ, *Pseudolinear Programming*, SIAM Review, 41 (1999), pp. 795–
710 805.
- 711 [26] T. KUNO, *A Branch-and-Bound Algorithm for Maximizing the Sum of Several Linear Ratios*,
712 Journal of Global Optimization, 22 (2002), pp. 155–174.
- 713 [27] D. LOVE AND G. BAYRAKSAN, *Phi-Divergence Constrained Ambiguous Stochastic Programs*
714 *for Data-Driven Optimization*, Technical report, Department of Integrated Systems Engineer-
715 ing, The Ohio State University, Columbus, Ohio, (2015).
- 716 [28] F. LUO AND S. MEHROTRA, *Decomposition Algorithm for Distributionally Robust Optimiza-*
717 *tion Using Wasserstein Metric*, arXiv preprint arXiv:1704.03920, (2017).
- 718 [29] F. LUO AND S. MEHROTRA, *Distributionally Robust Optimization with Decision Dependent*
719 *Ambiguity Sets*, arXiv preprint arXiv:1806.09215, (2018).
- 720 [30] F. LUO AND S. MEHROTRA, *Decomposition Algorithm for Distributionally Robust Optimiza-*
721 *tion using Wasserstein metric with an Application to a Class of Regression Models*, European
722 Journal of Operational Research, 278 (2019), pp. 20–35.
- 723 [31] G. P. MCCORMICK, *Computability of Global Solutions to Factorable Nonconvex Programs:*
724 *Part I - Convex Underestimating Problems*, Mathematical programming, 10 (1976), pp. 147–
725 175.
- 726 [32] S. MEHROTRA AND D. PAPP, *A Cutting Surface Algorithm for Semi-Infinite Convex Program-*
727 *ming with an Application to Moment Robust Optimization*, SIAM Journal on Optimization, 24
728 (2014), pp. 1670–1697.
- 729 [33] R. MISENER AND C. A. FLOUDAS, *GloMIQO: Global Mixed-Integer Quadratic Optimizer*,
730 Journal of Global Optimization, 57 (2013), pp. 3–50.
- 731 [34] Y. PEI AND D. ZHU, *Global Optimization Method for Maximizing the Sum of Difference*
732 *of Convex Functions Ratios over Nonconvex Region*, Journal of Applied Mathematics and

- Computing, 41 (2013), pp. 153–169.
- [35] N. T. H. PHUONG AND H. TUY, *A Unified Monotonic Approach to Generalized Linear Fractional Programming*, Journal of Global Optimization, 26 (2003), pp. 229–259.
- [36] H. RAHIMIAN, G. BAYRAKSAN, AND T. HOMEM-DE MELLO, *Identifying Effective Scenarios in Distributionally Robust Stochastic Programs with Total Variation Distance*, Mathematical Programming, 173 (2019), pp. 393–430.
- [37] H. RAHIMIAN AND S. MEHROTRA, *Distributionally Robust Optimization: A Review*, arXiv preprint arXiv:1908.05659, (2019).
- [38] M. RAO, *Cluster Analysis and Mathematical Programming*, Journal of the American Statistical Association, 66 (1971), pp. 622–626.
- [39] S. SCHAIBLE AND J. SHI, *Fractional Programming: The Sum-of-Ratios Case*, Optimization Methods and Software, 18 (2003), pp. 219–229.
- [40] A. SHAPIRO, *Distributionally Robust Stochastic Programming*, SIAM Journal on Optimization, 27 (2017), pp. 2258–2275.
- [41] A. SHAPIRO AND S. AHMED, *On a Class of Minimax Stochastic Programs*, SIAM Journal on Optimization, 14 (2004), pp. 1237–1249.
- [42] A. SHAPIRO AND A. KLEYWEGT, *Minimax Analysis of Stochastic Problems*, Optimization Methods and Software, 17 (2002), pp. 523–542.
- [43] B. K. SRIPERUMBUDUR AND G. R. LANCKRIET, *On the Convergence of the Concave–Convex Procedure*, in Neural Information Processing Systems, 2009, pp. 1759–1767.
- [44] J. STAHL, *Two New Methods for Solution of Hyperbolic Programming*, Publications of the Mathematical Institute of Hungarian Science, 9 (1964), pp. 743–754.
- [45] I. M. STANCU-MINASIAN, *Fractional Programming: Theory, Methods and Applications*, Springer, 2012.
- [46] R. A. STUBBS AND S. MEHROTRA, *A Branch-and-Cut Method for 0-1 Mixed Convex Programming*, Mathematical Programming, 86 (1999), pp. 515–532.
- [47] K. SWARUP, *Linear Fractional Functionals Programming*, Operations Research, 13 (1965), pp. 1029–1036.
- [48] Z. WANG, P. W. GLYNN, AND Y. YE, *Likelihood Robust Optimization for Data-Driven Problems*, Computational Management Science, 13 (2016), pp. 241–261.
- [49] W. WIESEMANN, D. KUHN, AND M. SIM, *Distributionally Robust Convex Optimization*, Operations Research, 62 (2014), pp. 1358–1376.
- [50] A. L. YUILLE AND A. RANGARAJAN, *The Concave-Convex Procedure*, Neural Computation, 15 (2003), pp. 915–936.

Appendix A. Proofs.

A.1. Proof of Proposition 3.1.

Proof. To prove the first part, we first show that $(x^*, \theta^*, \pi^*, \gamma^*)$ is feasible to (1.1). By $d_k^T x^* + \beta_k > 0$ and the constraints in (3.4), we have $(c_k^T x^* + \alpha_k)/(d_k^T x^* + \beta_k) \leq \gamma_k^*$ for all $k \in [K]$, which gives the feasibility of $(x^*, \theta^*, \pi^*, \gamma^*)$ to (1.1). Suppose that $(\hat{x}, \hat{\theta}, \hat{\pi}, \hat{\gamma})$ but not $(x^*, \theta^*, \pi^*, \gamma^*)$ is an optimal solution to (1.1) such that $\hat{\theta} < \theta^*$. Consider a solution $(\hat{x}, \hat{\theta}, \hat{\pi}, \bar{\gamma}, \bar{w}, \bar{v})$ that

$$\bar{\gamma}_k = \frac{c_k^T \hat{x} + \alpha_k}{d_k^T \hat{x} + \beta_k}, \quad \bar{w}_k = \frac{\bar{\gamma}_k + d_k^T \hat{x} + \beta_k}{2}, \quad \bar{v}_k = \frac{\bar{\gamma}_k - (d_k^T \hat{x} + \beta_k)}{2}, \quad k \in [K].$$

By construction, the solution $(\hat{x}, \hat{\theta}, \hat{\pi}, \bar{\gamma}, \bar{w}, \bar{v})$ satisfies (3.4) and $\bar{\gamma} \leq \hat{\gamma}$. By $H^T \bar{\pi} \geq \hat{\gamma}$ and $P\bar{\gamma} \leq \hat{\theta} \mathbf{1}_J$, we also have $H^T \bar{\pi} \geq \bar{\gamma}$ and $P\bar{\gamma} \leq \hat{\theta} \mathbf{1}_J$ since $\bar{\gamma} \leq \hat{\gamma}$ and $P \geq 0$, which implies that the solution $(\hat{x}, \hat{\theta}, \hat{\pi}, \bar{\gamma}, \bar{w}, \bar{v})$ is feasible to (3.6) with the objective value $\hat{\theta}$. This yields a contradiction to the fact that $(x^*, \theta^*, \pi^*, \gamma^*, w^*, v^*)$ is optimal to (3.6). So, $(x^*, \theta^*, \pi^*, \gamma^*)$ is optimal to (1.1).

For the second part, we first show the feasibility of $(x^*, \theta^*, \pi^*, \bar{\gamma}, \bar{w}, \bar{v})$ to (3.6). By construction, it satisfies (3.4). Also, by $H^T \pi^* \geq \gamma^*$ and $P\gamma^* \leq \theta^* \mathbf{1}_J$, we have $H^T \pi^* \geq \bar{\gamma}$ and $P\bar{\gamma} \leq \theta^* \mathbf{1}_J$ since $\bar{\gamma} \leq \gamma^*$ and $P \geq 0$. Therefore, $(x^*, \theta^*, \pi^*, \bar{\gamma}, \bar{w}, \bar{v})$ is feasible to (3.6). Suppose that $(x^*, \theta^*, \pi^*, \bar{\gamma}, \bar{w}, \bar{v})$ is not optimal to (3.6). Then, there exists an optimal solution $(\hat{x}, \hat{\theta}, \hat{\pi}, \hat{\gamma}, \hat{w}, \hat{v})$ to (3.6) such that $\hat{\theta} < \theta^*$. The argument in the first part indicates that the solution $(\hat{x}, \hat{\gamma}, \hat{w}, \hat{v})$ is feasible to (1.1) with the objective value $\hat{\theta}$. This leads to a contradiction to the fact that $(x^*, \theta^*, \pi^*, \gamma^*)$ is optimal to (1.1), which proves that $(x^*, \theta^*, \pi^*, \bar{\gamma}, \bar{w}, \bar{v})$ is optimal to (3.6). $\square$

A.2. Proof of Proposition 3.2.

Proof. The first part follows from

$$u(w) - w^2 = - \left(w - \frac{w^a + w^b}{2} \right)^2 + \frac{1}{4} (w^b - w^a)^2.$$

For the second part, the inequality trivially holds for $w = w^a$ and w^b . For $w \in (w^a, (w^a + w^b)/2]$, let g be a function such that $g(w) = (\hat{u}(w) - w^2)/(u(w) - w^2) = (2w - (w^a + w^b))/(2(w - w^b))$. Taking the derivative of g , we have $\partial g(w)/\partial w = (w^a - w^b)/(2(w - w^b)^2) < 0$. This results in $g(w) \leq \lim_{w \rightarrow w^a + 0} g(w) = 1/2$.

On the other hand, for $w \in [(w^a + w^b)/2, w^b]$, let $h(w) = (2w - (w^a + w^b))/(2(w - w^a))$. Since $\partial h(w)/\partial w = (w^a - w^b)/(2(w - w^a)^2) < 0$, we have $h(w) \leq \lim_{w \rightarrow w^b - 0} h(w) = 1/2$. This completes the proof. $\square$

A.3. Proof of Proposition 3.3.

Proof. By Proposition 3.1, without loss of generality, let $(x^*, \theta^*, \gamma^*, \pi^*, w^*, v^*)$ be an optimal solution to (3.6) satisfying (3.8). Since $w_k^* \in [w_k^m, w_k^M]$ holds for each $k \in [K]$, we have $(w_k^*)^2 \leq u(w_k^*; W_k)$, which leads to $c_k^T x^* + \alpha_k + (v_k^*)^2 \leq (w_k^*)^2 \leq u(w_k^*; W_k)$ for every $k \in [K]$. Therefore, $(x^*, \theta^*, \gamma^*, \pi^*, w^*, v^*)$ is feasible to (3.10) with the objective value of $\theta^* = \vartheta^*$, which completes the proof. $\square$

Appendix B. Disjunctive cuts. To solve the mixed binary program (3.13) in the branch and cut framework, we introduce a separation problem which generates a hyperplane that cuts off a fractional solution from the convex hull of the feasible region. Our derivation follows the variable transformation technique introduced in (46).

Let $(\bar{x}, \bar{\theta}, \bar{\gamma}, \bar{\pi}, \bar{w}, \bar{v}, \bar{\lambda}, \bar{\mu})$ be a fractional solution obtained by solving a relaxation problem of (3.13). By (A1)-(A5), (1.1) and thus (3.13) have a finite optimum. Let $(x^*, \theta^*, \gamma^*, \pi^*, w^*, v^*, \lambda^*, \mu^*)$ be an optimal solution to (3.13). Then, there exists a constant L satisfying

$$|\theta^*| \leq L, \quad \max \{\|x^*\|_\infty, \|\gamma^*\|_\infty, \|\pi^*\|_\infty, \|w^*\|_\infty, \|v^*\|_\infty\} \leq L, \quad \max_{j \in [N_k], k \in [K]} |\lambda_{kj}^*| \leq L,$$

$$\|Ax^* - b\|_\infty \leq L, \quad -\|x^*\|_\infty \leq L, \quad \max_{k \in [K]} \sum_{j=1}^{N_k} \lambda_{kj}^* (w_k^j)^2 - (c_k^T x^* + \alpha_k + (v_k^*)^2) \leq L.$$

Without loss of generality, we can impose an upper bound L for each element of feasible solutions and constraints in (3.13). With the bound constraints, the feasible region of (3.13) can be represented as the union of $\prod_{k=1}^K (N_k - 1)$ sets of the form

$$C^{j_1 j_2 \dots j_K} := \left\{ (x, \theta, \gamma, \pi, w, v, \lambda) \left| \begin{array}{l} |\theta| \leq L, \|x\|_\infty \leq L, \|\pi\|_\infty \leq L, \\ \|\gamma\|_\infty \leq L, \|w\|_\infty \leq L, \|v\|_\infty \leq L, \\ \|Ax - b\|_\infty \leq L, -\|x\|_\infty \leq L, \\ |\lambda_{kj}| \leq L, j \in [N_k], k \in [K], \\ \sum_{j=1}^{N_k} \lambda_{kj} (w_k^j)^2 - (c_k^T x + \alpha_k + v_k^2) \leq L, k \in [K], \\ c_k^T x + \alpha_k + v_k^2 \leq \sum_{j=1}^{N_k} \lambda_{kj} (w_k^j)^2, k \in [K], \\ \mu_k^{j_k} = \mu_k^{j_k+1} = 1, k \in [K], \\ (w, \lambda_k, \mu_k) \in \mathcal{H}_k, k \in [K], \\ (x, \theta, \gamma, \pi, w, v) \in \mathcal{S}. \end{array} \right. \right\}$$

where $j_k \in [N_k - 1]$ and $k \in [K]$. Note that $C^{j_1 j_2 \dots j_K}$ represents a set of points satisfying $w_k \in [w_k^{j_k}, w_k^{j_k+1}]$ and $c_k^T x + \alpha_k + v_k^2 \leq (w_k^{j_k} + w_k^{j_k+1})w_k - w_k^{j_k} w_k^{j_k+1}$ for $k \in [K]$.

To construct the convex hull of $C^{j_1 j_2 \dots j_K}$, we use the variable transformation technique in (46). Rather than writing a convex combination $z = (x, \theta, \gamma, \pi, w, v, \lambda)$ as

$$z = \sum_{j_1=1}^{N_1-1} \sum_{j_2=1}^{N_2-1} \dots \sum_{j_K=1}^{N_K-1} \eta^{j_1 j_2 \dots j_K} z^{j_1 j_2 \dots j_K}$$

where

$$\sum_{j_1=1}^{N_1-1} \sum_{j_2=1}^{N_2-1} \dots \sum_{j_K=1}^{N_K-1} \eta^{j_1 j_2 \dots j_K} = 1, \quad \eta^{j_1 j_2 \dots j_K} \geq 0, \quad z^{j_1 j_2 \dots j_K} \in C^{j_1 j_2 \dots j_K}, \quad j_k \in [N_k - 1], \quad k \in [K],$$

using (46, Theorem 2), we represent z as

$$z = \sum_{j_1=1}^{N_1-1} \sum_{j_2=1}^{N_2-1} \dots \sum_{j_K=1}^{N_K-1} \hat{z}^{j_1 j_2 \dots j_K}$$

where

$$(\hat{z}^{j_1 j_2 \dots j_K}, \eta^{j_1 j_2 \dots j_K}) \in \hat{C}^{j_1 j_2 \dots j_K}, \quad j_k \in [N_k - 1], \quad k \in [K], \quad \sum_{j_1=1}^{N_1-1} \sum_{j_2=1}^{N_2-1} \dots \sum_{j_K=1}^{N_K-1} \eta^{j_1 j_2 \dots j_K} = 1$$

819 and

$$820 \quad \hat{C}^{j_1 j_2 \dots j_K} := \left(\hat{z}, \eta \right) \left\{ \begin{array}{l} |\hat{\theta}| \leq \eta L, \|\hat{x}\|_\infty \leq \eta L, \|\hat{\pi}\|_\infty \leq \eta L, \\ \|\hat{\gamma}\|_\infty \leq \eta L, \|\hat{w}\|_\infty \leq \eta L, \|\hat{v}\|_\infty \leq \eta L, \\ \|A\hat{x} - b\eta\|_\infty \leq \eta L, -\|\hat{x}\|_\infty \leq \eta L, |\hat{\lambda}_{kj}| \leq \eta L, j \in [N_k], k \in [K], \\ A\hat{x} \leq b\eta, \hat{x} \geq 0, f^T \hat{\pi} \leq \hat{\theta}, H^T \hat{\pi} \geq \hat{\gamma}, P\hat{\gamma} \leq \hat{\theta} \mathbf{1}_J, \\ \hat{\gamma}_k - (d_k^T \hat{x} + \beta_k \eta) = 2\hat{v}_k, \hat{\gamma}_k + (d_k^T \hat{x} + \beta_k \eta) = 2\hat{w}_k, k \in [K], \\ \hat{w}_k = \sum_{j=1}^{N_k} \hat{\lambda}_{kj} (w_k^j)^2, c_k^T \hat{x} + \alpha_k \eta + \hat{v}_k^2 / \eta \leq \sum_{j=1}^{N_k} \hat{\lambda}_{kj} (w_k^j)^2, k \in [K], \\ 0 \leq \hat{\lambda}_{kj} \leq \hat{\mu}_{kj}, j \in [N_k], \sum_{j=1}^{N_k} \hat{\lambda}_{kj} = \eta, k \in [K], \\ \hat{\mu}_{kj_k} = \hat{\mu}_{kj_k+1} = \eta, \hat{\mu}_{ki_k} = 0, i_k \neq j_k, j_k + 1, k \in [K], \\ \hat{z} = (\hat{x}, \hat{\theta}, \hat{\gamma}, \hat{\pi}, \hat{w}, \hat{v}, \hat{\lambda}), \eta \geq 0. \end{array} \right\}.$$

821

822 In the above, $\eta^{j_1 j_2 \dots j_K}$ represents a coefficient in a convex combination and $\hat{z}^{j_1 j_2 \dots j_K}$ is a coefficient
 823 multiplied quantity such that $\hat{z}^{j_1 j_2 \dots j_K} / \eta^{j_1 j_2 \dots j_K} \in C^{j_1 j_2 \dots j_K}$ for $\eta^{j_1 j_2 \dots j_K} > 0$.

824 To cut off a fractional solution $\bar{z} = (\bar{x}, \bar{\theta}, \bar{\gamma}, \bar{\pi}, \bar{w}, \bar{v}, \bar{\lambda})$, we consider the following separation
 825 problem

$$\begin{aligned} \min_{\bar{z}} \quad & f(\bar{z}) = \|\bar{z} - \bar{z}\| \\ \text{subject to} \quad & \bar{z} = \sum_{j_1=1}^{N_1-1} \sum_{j_2=1}^{N_2-1} \dots \sum_{j_K=1}^{N_K-1} \hat{z}^{j_1 j_2 \dots j_K}, \\ (B.1) \quad & \sum_{j_1=1}^{N_1-1} \sum_{j_2=1}^{N_2-1} \dots \sum_{j_K=1}^{N_K-1} \eta^{j_1 j_2 \dots j_K} = 1, \\ & (\hat{z}^{j_1 j_2 \dots j_K}, \eta^{j_1 j_2 \dots j_K}) \in C^{j_1 j_2 \dots j_K}, j_k \in [N_k - 1], k \in [K]. \end{aligned}$$

826

827 If an optimal solution z^* satisfies $f(z^*) > 0$, then by (46, Theorem 3), the inequality $\partial f(z^*)^T (z -$
 828 $\bar{z}) \geq 0$ constructed by the subgradient $\partial f(z^*)$ and the fractional solution $\bar{z}$ is a valid inequality in
 829 z and cuts off $\bar{z}$. For more information, see (46).