An Algorithm for Stochastic Convex-Concave Fractional Programs with Applications to Production Efficiency and Equitable Resource Allocation

Shibshankar Dey[a], Cheolmin Kim[a], Sanjay Mehrotra[a,*]

[a]*Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 60208*

Abstract

We propose an algorithm to solve convex and concave fractional programs and their stochastic counterparts in a common framework. Our approach is based on a novel reformulation that involves differences of square terms in the constraints, and subsequent employment of piecewise-linear approximations of the concave terms. Using the branch-and-bound (B&B) framework, our algorithm adaptively refines the piecewise-linear approximations and iteratively solves convex approximation problems. The convergence analysis provides a bound on the optimality gap as a function of approximation errors. Based on this bound, we prove that the proposed B&B algorithm terminates in a finite number of iterations and the worst-case bound to obtain an ϵ-optimal solution reciprocally depends on the square root of ϵ. Numerical experiments on Cobb-Douglas production efficiency and equitable resource allocation problems support that the algorithm efficiently finds a highly accurate solution while significantly outperforming the benchmark algorithms for all the small size problem instances solved. A modified branching strategy that takes the advantage of non-linearity in convex functions further improves the performance. Results are also discussed when solving a dual reformulation and using a cutting surface algorithm to solve

*Corresponding author

Email addresses: `shibshankardey2025@u.northwestern.edu` (Shibshankar Dey), `cheolminkim2019@u.northwestern.edu` (Cheolmin Kim), `mehrotra@northwestern.edu` (Sanjay Mehrotra)

distributionally robust counterpart of the Cobb-Douglas example models.

Keywords: Fractional programming, Second order cone approximation, Branch and bound algorithm, Stochastic production efficiency problem, Equitable resource allocation

1. Introduction

Let us consider fractional optimization models of the form

$$\min_{x\in\mathcal{X}} \quad \sum_{k=1}^{K} p_k \frac{f_k(x)}{g_k(x)}, \tag{1}$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ and $p_k \geq 0$ for every $k \in [K] := \{1, 2, \cdots, K\}$. If the vector $p \in \mathbb{R}^K_+$ satisfies $\sum_{k=1}^K p_k = 1$, the model can be viewed as a stochastic fractional program with finite support. Moreover, by taking $p_k = 1/K$ the model can be understood as a sample average approximation of a stochastic fractional program:

$$\min_{x\in\mathcal{X}} \quad \bar{h}(x) := \mathbb{E}_{\mathbb{P}}\left[\frac{\tilde{f}(x)}{\tilde{g}(x)}\right], \tag{2}$$

where the expectation is defined using a measure from the probability space $(\Omega, \mathfrak{B}, \mathbb{P})$, and $\tilde{f}(\cdot)$, $\tilde{g}(\cdot)$ denote functions with random parameters following a probability distribution $\mathbb{P}$.

We study two optimization models of (1). In the first model, f_k are convex and g_k are concave, and $f_k(x) > 0$ and $g_k(x) > 0$ for all $x \in \mathcal{X}$ and $k \in [K]$. This model is called the *convex fractional program*. The second model is the *concave fractional program*, which has the form of $\max_{x\in\mathcal{X}} \sum_{k=1}^K p_k \frac{f'_k(x)}{g'_k(x)}$ where f'_k are concave and g'_k are convex, and $f'_k(x) > 0$ and $g'_k(x) > 0$ for all $x \in \mathcal{X}$ and $k \in [K]$. Converting it to a minimization problem, we can represent this model in the form of (1). These two models have been studied independently in the literature. In this work, we provide a unified framework that covers not only the convex and concave fractional programs but also their distributionally robust counterparts. We give two motivating examples below.

1.1. Motivating Examples

Our consideration of (1) is motivated from the following two applications. The first one is the stochastic version of the equitable resource allocation problem, which aims to allocate resources to entities in an equitable manner. This allocation model is useful when the available resources from the suppliers are in short supply. The second application model is a stochastic generalization of the classical Cobb-Douglas model for measuring production efficiency.

1.1.1. Stochastic Equitable Resource Allocation Problem

Suppose that we have m suppliers and n customers. Let r_i be the amount of resources supplier i can provide and d_j be the requirement of customer j. Let b_{ij} be the benefit each unit resource from supplier i brings to customer j. The decision variable x_{ij} allocates resources from supplier i to customer j. A classical model, without equity considerations, maximizes total benefits by solving

$$\begin{aligned} \theta^* := \max \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} \le r_i, \sum_{i=1}^{m} b_{ij} x_{ij} \le d_j, \, x_{ij} \ge 0, \, i \in [m], \, j \in [n]. \end{aligned} \tag{3}$$

However, a solution to (3) may lead to unfair allocation of available resources to the customers. An equitable resource allocation model balances benefit maximization with allocation equity. The objective function in the equitable resource allocation model minimizes an equity-based objective function, while ensuring that the total benefits from the allocation do not fall below a certain threshold $\delta\theta^*$, where $\delta \in [0, 1]$ and θ^* is the maximum value in (3). With this consideration, the equitable resource allocation problem is formulated as

$$\begin{aligned}
\min \quad & \sum_{j=1}^{n} \left| \frac{\sum_{i=1}^{m} b_{ij} x_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij} x_{ij}} - \frac{1}{n} \right| \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} \le r_i,\ \sum_{i=1}^{m} b_{ij} x_{ij} \le d_j,\ x_{ij} \ge 0,\ i \in [m],\ j \in [n], \\
& \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij} x_{ij} \ge \delta \theta^*.
\end{aligned} \tag{4}$$

This model allocates the resources so that each customer achieves a nearly equal share of the total benefit. In problem (4), $f(x) = \sum_{j=1}^{n} | \sum_{i=1}^{m} b_{ij} x_{ij} - \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij} x_{ij} | / n$ and $g(x) = | \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij} x_{ij} |$. $\sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij} x_{ij} \ge 0$ holds for all $x \ge 0$, since $b_{ij} \ge 0$, thus $g(x) := \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij} x_{ij}$ is a linear function and the model in (4) is a convex fractional program.

In the above model if parameters b, r, d are random and they follow a discrete probability distribution, we have the stochastic equitable resource allocation problem:

$$\begin{aligned}
\min \quad & \sum_{k=1}^{K} p_k \sum_{j=1}^{n} \left| \frac{\sum_{i=1}^{m} b_{ij}^k x_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij}^k x_{ij}} - \frac{1}{n} \right| \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} \le r_i^k,\ \sum_{i=1}^{m} b_{ij}^k x_{ij} \le d_j^k,\ x_{ij} \ge 0,\ i \in [m],\ j \in [n],\ k \in [K] \\
& \sum_{k=1}^{K} p_k \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij}^k x_{ij} \ge \delta \overline{\theta}^*.
\end{aligned} \tag{5}$$

where K is the total number of scenarios, $p_k \ge 0$ is the probability of scenario k, $\sum_{k=1}^{K} p_k = 1$, all scenario specific parameters are superscripted with k, δ has the same interpretation as in (4) and $\overline{\theta}^*$ is the optimal value to stochastic variant of (3) as follows:

$$\begin{aligned}
\overline{\theta}^* := \max \quad & \sum_{k=1}^{K} p_k \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij}^k x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} \le r_i^k,\ \sum_{i=1}^{m} b_{ij}^k x_{ij} \le d_j^k,\ x_{ij} \ge 0,\ i \in [m],\ j \in [n],\ k \in [K].
\end{aligned} \tag{6}$$

1.1.2. Stochastic Cobb-Douglas Production Efficiency Problem

The Cobb-Douglas production function (Cobb & Douglas, 1928) aggregates economy-wide information. Historically, it was the first production function that was estimated and used for analysis. The analysis of this function resulted in a landmark step in modeling macroeconomics from the microeconomics perspective (Filipe & Adams, 2005). The Cobb-Douglas model in (Bradley & Frey Jr, 1974) uses the profit function of a firm as $f(x) = a_0 \prod_{i=1}^n x_i^{a_i}$ where x_j are production factors, and $a_1, a_2, \cdots, a_n$ are nonnegative parameters such that $\sum_{i=1}^n a_i = 1$. Due to this constraint on $a_1, a_2, \cdots, a_n$, the function f is concave in x. The set $\mathcal{X} \in \mathbb{R}_{++}^n$ describes the domain of production factors. The total cost is a linear function of production factors and it is given by $g(x) = \sum_{i=1}^n c_i x_i + c_0$. The production efficiency problem is formulated as

$$\max_{x \in \mathcal{X}} \quad \frac{a_0 \prod_{i=1}^n x_i^{a_i}}{\sum_{i=1}^n c_i x_i + c_0}. \tag{7}$$

In this model, we may assume that the parameters follow a probability distribution $\mathbb{P}$. This results in the stochastic programming generalization of the Cobb-Douglas model. More generally, assuming that the model parameters a and c follow an unknown probability distribution $\mathbb{P}$, which is contained in a set of probability distributions, called an ambiguity set, $\mathcal{D}$, a distributionally robust Cobb-Douglas production efficiency model can be formulated as:

$$\max_{x \in \mathcal{X}} \min_{\mathbb{P} \in \mathcal{D}} \quad \mathbb{E}_{(\tilde{a}, \tilde{c}) \sim \mathbb{P}} \left[\frac{\tilde{a}_0 \prod_{i=1}^n x_j^{\tilde{a}_i}}{\sum_{i=1}^n \tilde{c}_i x_i + \tilde{c}_0} \right]. \tag{8}$$

The model (8) specializes to a concave fractional program if the set $\mathcal{D}$ is a singleton and its element $\mathbb{P}$ has finite support.

1.1.3. Other Applications

While the development of solution approaches in this paper is motivated from (4) and (8), the developed methodology can be applied to other applications such as those arising in information theory (Meister & Oettli, 1967; Aggarwal & Sharma, 1970), cluster analysis (Rao, 1971), portfolio investment problems

(Ziemba et al., 2013) and inventory problems (Hodgson & Lowe, 1982). For more applications, see Stancu-Minasian (1997).

1.2. Contributions

This paper studies convex and concave fractional programming problems and their stochastic counterparts in a common framework. This is a non-convex optimization problem. We reformulate the problem through piecewise linear approximation by using the concept introduced in Kim & Mehrotra (2021) for stochastic fractional linear programs. We show that the sample average approximation (SAA) of stochastic convex and concave fractional programs converge to its true optima with increasing sample size, and also provide a result similar to the central limit theorem. An algorithm is developed that adaptively refines this piecewise-linear approximation by dividing a hyper-rectangle and solving a convex approximation problem for each sub-hyper-rectangle to update the lower bound and the incumbent solution. The basic idea of approximating the difference of quadratic functions using a piecewise-linear approximation was introduced in Kim & Mehrotra (2021) in the context of linear fractional programming and its stochastic counterparts. This work generalizes its applicability to a much broader setting. A convergence analysis shows that the algorithm attains an ϵ-optimal solution after a finite number of iterations. Specifically, the worst-case bound for the number of iterations is in the order of $\mathcal{O}(1/\sqrt{\epsilon})$. This is an improvement of $\mathcal{O}(1/\sqrt{\epsilon})$ over the previous results, and indicates its efficiency in finding a more accurate solution.

The experimental results show that with a 12-hour time limit the proposed branch-and-bound algorithm outperforms benchmark algorithms on test instances for both problems. For 10-scenario stochastic resource allocation problem, the proposed algorithm achieves given relative optimality gap within the time limit for majority of the instances. Two to four digit accuracy is achieved in the remaining instances. However, previously known benchmark algorithms cannot achieve any digit accuracy. For the Cobb-Douglas instances of dimension up to 15, the proposed algorithm attains the desired solution accuracy for all cases while benchmark algorithms never attain this accuracy when used with a 12 hour time limit. A novel LP-relaxation based branching strategy further improves the efficiency by about 50% on average.

We discuss two solution approaches for the distributionally robust formulations. They are based on a dual reformulation and the cutting surface algorithm in Section 6. The dual approach is applicable when it is possible to dualize the ambiguity set $\mathcal{D}$ without an optimality gap and the introduction of dual variables do not change the structure of the model formulation, as in the case when $\mathcal{D}$ has a polyhedral description. The cutting surface approach allows the use of a general convex set when specifying $\mathcal{D}$. For distributionally robust Cobb-Douglas production efficiency problems, these solution approaches attain the desired solution accuracy with a little extra computation if nominal data is uniformly distributed. However, the problems become more challenging to solve if the nominal data follows a skewed distribution.

2. Literature Review

2.1. Algorithms for Convex Fractional Program

Konno et al. (1994) considers a generalized convex multiplicative programming problem which minimizes $r(x) + \sum_{k=1}^{K} f_k(x)h_k(x)$ subject to a compact and convex set $\mathcal{X}$ where $f_k(x) > 0$, $h_k(x) > 0$, $k \in [K]$ and $r(x)$ are convex functions. This optimization problem specializes to the convex-concave fractional program when $r(x) = 0$ and $h_k(x) = 1/g_k(x)$ for concave g_k. This work presents an outer approximation algorithm that solves a sequence of approximation problems. The approximation problems are concave minimization problems and the feasible region is successively refined through linear cuts. The algorithm attains ϵ-optimal solution after a finite number of iterations.

Freund & Jarre (2001) and Benson (2001) present branch and bound algorithms to solve the convex-concave fractional program. In Freund & Jarre (2001), a K-dimensional hyper-rectangle containing the Cartesian product of the ranges of g_k are branched. For each hyper-rectangle, they solve a convex optimization problem using an interior point method, and use the resulting Lagrange multipliers to obtain a linear function, which results in lower bounds for two sub-hyper-rectangles. On the other hand, Benson (2001) branches a $2K$-dimensional hyper-rectangle, which contains the Cartesian product of the ranges of f_k and g_k. For each hyper-rectangle, it solves two different convex optimization problems to obtain a lower bound. The branch-and-bound algorithm in Benson (2001) is shown to obtain an ϵ-optimal solution in a finite number of iterations.

2.2. Algorithms for Concave Fractional Program

Dur et al. (2001), Benson (2002a), and Benson (2002b) present branch and bound algorithms to solve the concave-convex fractional program. Dur et al. (2001) introduces K auxiliary variables for fractional terms, and successively branches them and solves convex approximation problems in a branch-and-bound framework. In Benson (2002a), K auxiliary variables are introduced for reciprocals of g_k. By branching them and solving convex approximation problems, it finds an ϵ-optimal solution. The convergence result states that either the algorithm terminates after a finite number of iterations or every accumulation point of a sequence of incumbent solutions is an optimal solution. On the other hand, Benson (2002b) introduces $2K$ auxiliary variables for f_k and g_k. K auxiliary variables for g_k are branched and convex relaxation problems are derived using the McCormick envelope (McCormick, 1976). The convergence result in Benson (2002b) is similar to the one given for the algorithm in Benson (2002a).

In additional literature, Gruzdeva & Strekalovsky (2018) developed a solution approach for general functions in the fractional form. This approach can be adapted to convex-concave fractional programs. However, it does not provide a convergence and performance guarantee. The algorithm in Jiao & Liu (2017) for the sum of ratios problem is limited to quadratic functions in the numerator and denominator. Hu et al. (2019) propose an incremental quasi-subgradient method to solve the sum of convex-concave ratio problem. A variant of their method randomly chooses an element from the summation terms. This algorithm is shown to probabilistically converge. However, the analysis makes a strong homogeneity assumption that all the fractional terms in the objective, if optimized individually, have at least one common optimal point.

We also observe that the Cobb-Douglas production efficiency problem can be treated as a variant of a general geometric program. Algorithms for general geometric programming that are developed in the recent literature such as Wang & Liang (2005) can thus also be leveraged to solve this specific problem.

3. Convex Approximations

In this section, we propose a general framework that covers convex and concave fractional programs and their stochastic counterparts as special cases. We introduce a reformulation that involves difference-of-convex constraints and present the idea of piecewise-linear approximation.

3.1. A General Framework

Let us consider a fractional program of the form

$$\begin{aligned} \min_{\theta,x,c,d,\gamma,\pi} \quad & \theta \\ \text{s.t.} \quad & f_k(x) \le c_k \le \alpha_k,\ \sigma\beta_k \le \sigma d_k \le g_k(x),\ \frac{c_k}{d_k} \le \gamma_k,\ k \in [K], \\ & f^T\pi \le \theta,\ H^T\pi \ge \gamma,\ P\gamma \le \theta\mathbb{1}_J, x \in \mathcal{X},\ \theta \in \mathbb{R},\ \gamma \in \mathbb{R}^K,\ \pi \in \mathbb{R}^L, \end{aligned} \tag{9}$$

where $x \in \mathbb{R}^n$, $\mathcal{X} \subseteq \mathbb{R}^n$, and f_k and g_k are numerator and denominator functions, respectively. The functions f_k and g_k are bounded by variables c_k and d_k, which have an upper bound through constants α_k and β_k. The fractional terms c_k/d_k, are the only non-convex terms in this model. The vector γ affects the objective value θ through either $f^T\pi \le \theta$, $H^T\pi \ge \gamma$ or $P\gamma \le \theta\mathbb{1}_J$ where J is the number of rows of probability matrix P. In stochastic programs (1), it is only a row vector of $p_1, p_2, \ldots, p_K$. For distributionally robust counterpart, if the cutting surface method is used, matrix P represents the set of probability distributions generated in the algorithm from sequentially adding the probability cuts as row vectors to the matrix (see Section 6.2 for details).

We make the following assumptions throughout the paper.

(A1) $\mathcal{X} \subset \mathbb{R}^n$ is a non-empty compact and convex set.

(A2) f_k are convex functions and g_k are concave functions.

(A3) P is a non-negative matrix in $\mathbb{R}^{J\times K}$.

(A4) $\mathcal{P} := \{p\,|Hp = f, p \geq 0\}$ is a non-empty polytope in $\mathbb{R}^K$.

(A5) For $\sigma = 1$ we assume that $0 < f_k(x) < \infty$ and $0 < \delta_g \leq g_k(x) < \infty$ for some positive constant δ_g for all $x \in \mathcal{X}, k \in [K]$. $\max_{x\in\mathcal{X}} f_k(x) \leq \alpha_k < \infty$ and $0 < \beta_k \leq \min_{x\in\mathcal{X}} g_k(x)$ for all $k \in [K]$.

(A6) For $\sigma = -1$ we assume that $-\infty < f_k(x) < 0$ and $-\infty < g_k(x) \leq -\delta_g < 0$ for some positive constant δ_g for all $x \in \mathcal{X}$ and $k \in [K]$; $\max_{x\in\mathcal{X}} f_k(x) \leq \alpha_k \leq 0$ and $\max_{x\in\mathcal{X}} -g_k(x) \leq \beta_k < \infty$ for all $k \in [K]$.

3.1.1. Convex Fractional Program

Let $\alpha_k = \max_{x\in\mathcal{X}} f_k(x)$, $\beta_k = \min_{x\in\mathcal{X}} g_k(x)$. Since $f_k(x) > 0$, $g_k(x) > 0$ for all $x \in \mathcal{X}$ and $k \in [K]$, we can write the convex fractional program as

$$\begin{aligned} \min \quad & \theta \\ \text{s.t.} \quad & f_k(x) \leq c_k \leq \alpha_k,\ \beta_k \leq d_k \leq g_k(x),\ \frac{c_k}{d_k} \leq \gamma_k,\ k \in [K], \\ & p^T\gamma \leq \theta,\ x \in \mathcal{X},\ \theta \in \mathbb{R},\ \gamma \in \mathbb{R}^K. \end{aligned} \tag{10}$$

This has the form of (9) with $\sigma = 1$. The constants α_k are computed in a preprocessing step. It is problem specific, and discussed further in Sections 7.

3.1.2. Concave Fractional Program

The concave fractional program has the form of $\max_{x\in\mathcal{X}} \sum_{k=1}^{K} p_k \frac{f'_k(x)}{g'_k(x)}$ where f'_k are concave and g'_k are convex for $k \in [K]$, and $f'_k(x) > 0$ and $g'_k(x) > 0$ for all $x \in \mathcal{X}$ and $k \in [K]$. Let $\alpha_k = \max_{x\in\mathcal{X}} -f'_k(x)$, $\beta_k = \max_{x\in\mathcal{X}} g'_k(x)$. Since

$f'_k(x) > 0$ and $g'_k(x) > 0$ for all $x \in \mathcal{X}$ and $k \in [K]$, we can write it as

$$\begin{aligned} \min \quad & \theta \\ \text{s.t.} \quad & -f'_k(x) \le c_k \le \alpha_k,\ g'_k(x) \le d_k \le \beta_k,\ \frac{c_k}{d_k} \le \gamma_k,\ k \in [K], \\ & p^T\gamma \le \theta,\ x \in \mathcal{X},\ \theta \in \mathbb{R},\ \gamma \in \mathbb{R}^K. \end{aligned} \tag{11}$$

Note that the concave fractional program (11) is a special case of (9) with $\sigma := -1, f_k := -f'_k, g_k := -g'_k$.

3.1.3. Distributionally Robust Fractional Program

If the ambiguity set is polyhedral or convex, we can write a reformulation or a subproblem of a distributionally robust convex or concave fractional program in the form of (9) (see Section 6).

3.2. Convergence of SAA of Stochastic Fractional Program

We use the general theory from Shapiro (1991) to give a convergence result of SAA in our case. The SAA convergence results in Shapiro (1991) rely on certain assumptions on a function parameterized by random parameters. We state these results below.

Theorem 3.1. *[Theorem 3.2, Shapiro (1991)] Let $\mathcal{X}$ be compact, $\{h_K\}$ be a sequence of random elements in Banach Space $\mathcal{B}(\mathcal{X})$, $\bar{h} \in \mathcal{B}(\mathcal{X})$ and $\mathcal{X}^*(\bar{h})$ is the set of minimizer of $\bar{h}(x)$ over $\mathcal{X}$. Suppose that $\sqrt{K}(h_K - \bar{h})$ converges in distribution to a random element Z of $\mathcal{B}(\mathcal{X})$. Let $\phi^* := \min_{x\in\mathcal{X}} \bar{h}(x)$ and $\hat{\phi}_K$ be the objective value of a sample average approximation problem from K samples. Then $\sqrt{K}\,(\hat{\phi}_K - \phi^*) \xrightarrow{D} \min_{x\in\mathcal{X}^*(\bar{h})} Z(x)$ too. In particular, if $\bar{h}(x)$ attains its minimum over $\mathcal{X}$ at a unique point x^* then $\sqrt{K}\,(\hat{\phi}_K - \phi^*)$ converges in distribution to $Z(x^*)$.*

Theorem 3.2. *[Theorem 3.3, Shapiro (1991)]. Suppose that $\bar{h}(x)$ has a unique minimizer x^* over $\mathcal{X}$. Assume that the following three conditions are satisfied: (a) the function $h(x,\cdot)$ is measurable for every $x \in \mathcal{X}$, (b) the expectation $\mathbb{E}\left[h(x,w)^2\right]$ is finite for some point $x^* \in \mathcal{X}$, (c) there exists a function*

$L : \Omega \to \mathbb{R}$ *such that* $\mathbb{E}\left[L(w)^2\right]$ *is finite and that* $|h(x,\omega) - h(y,\omega)| \le L(\omega)\,\|x - y\| \quad \forall x, y \in \mathcal{X}$. *Then* $\sqrt{K}\,(\hat{\phi}_K - \phi^*) \xrightarrow{D} \mathcal{N}(0,\sigma)$ *with* $\sigma^2 = \mathbb{E}\left[h(x^*,w)^2\right] - \left(\mathbb{E}\left[h(x^*,w)\right]\right)^2$.

We make the following additional assumptions for our SAA convergence analysis of the stochastic convex fractional program (2) ($\sigma = 1$ case). Analogous assumptions can be made to develop a similar proof for the stochastic concave fractional programs ($\sigma = -1$).

(A7) The sample space Ω is compact.

(A8) *For all* $\omega \in \Omega$, the functions $f(x,\omega)$, $g(x,\omega)$ are bounded, and $0 < \delta_g \le g(x,\omega)$. They satisfy $|f(x,\omega) - f(y,\omega)| \le L_f(\omega)\,\|x - y\|$, $|g(x,\omega) - g(y,\omega)| \le L_g(\omega)\,\|x - y\| \quad \forall x, y \in \mathcal{X}$, $\forall \omega \in \Omega$, for some $L_f(\cdot), L_g(\cdot) : \Omega \to \mathbb{R}$. Moreover, $M_f := \max_{\omega\in\Omega} L_f(\omega)$, $M_g := \max_{\omega\in\Omega} L_g(\omega)$ exist.

A consequence of Assumption A7 is that $f(x,\omega)$, and $g(x,\omega)$ are $\mathcal{L}_2$ Lebesgue integrable[1] for all $x \in \mathcal{X}$. Moreover, $L_f(\omega), L_g(\omega)$ are also $\mathcal{L}_2$-Lebesgue integrable for all $x \in \mathcal{X}$. The following lemma is needed to use Theorems 3.2-3.2 in our context. It shows that under Assumptions A7 and A8 sufficient conditions in Theorems 3.2-3.2 are satisfied.

Lemma 3.1. *Let* $h(x,\omega) = f(x,\omega)/g(x,\omega)$. *Then under Assumptions (A7)-(A8) the following holds:*

a) *$h(x,\omega)$ is $\mathcal{L}_2$ Lebesgue integrable for some $x^0 \in \mathcal{X}$, i.e., $\mathbb{E}[h(x^0,\omega)^2] < \infty$.*

b) *There exists a Lipschitz function $L_h(\cdot) : \Omega \to \mathbb{R}$ such that $\mathbb{E}[L_h(\omega)^2]$ is finite and $|h(x,\omega) - h(y,\omega)| \le L_h(\omega)\|x - y\|$, $L_h(\omega)$ is finite for all $x, y \in \mathcal{X}$ and $\omega \in \Omega$.*

Proof. Part (a) follows because $\mathbb{E}[h(x^0,\omega)^2] = \mathbb{E}[\frac{f(x^0,\omega)^2}{g(x^0,\omega)^2}] \le \frac{\mathbb{E}[f(x^0,\omega)^2]}{\delta_g^2}$, $\delta_g > 0$, $\mathbb{E}[f(x^0,\omega)^2]$ is finite for all $\omega \in \Omega$ (Assumption A8). We prove part (b) by contradiction. The claim in part (b) can fail in two ways: (i) $\nexists$ finite $L_h(\hat{\omega})$

[1] A measurable function $f : \Omega \to \mathbb{R}$ is called $\mathcal{L}_2$-Lebesgue integrable if $\int_\Omega |f|^2\, d\mathbb{P} < \infty$.

such that $|h(\hat{x},\hat{\omega}) - h(\hat{y},\hat{\omega})| \le L_h(\hat{\omega})\|\hat{x}-\hat{y}\|$ holds for some $\hat{\omega} \in \Omega$, $\hat{x},\hat{y} \in \mathcal{X}$; or (ii) although (i) holds but $\mathbb{E}[L_h(\omega)^2]$ is not finite. Assume that (i) fails to hold. It implies that for $\hat{x} \neq \hat{y} \in \mathcal{X}$

$$
\begin{aligned}
&\frac{1}{\|\hat{x}-\hat{y}\|}\left|\frac{f(\hat{x},\hat{\omega})}{g(\hat{x},\hat{\omega})} - \frac{f(\hat{y},\hat{\omega})}{g(\hat{y},\hat{\omega})}\right| \text{ is unbounded above} \\
\Rightarrow &\frac{1}{\|\hat{x}-\hat{y}\|}\quad\left|\frac{f(\hat{x},\hat{\omega})g(\hat{y},\hat{\omega}) - f(\hat{y},\hat{\omega})g(\hat{x},\hat{\omega})}{g(\hat{x},\hat{\omega})g(\hat{y},\hat{\omega})}\right| \text{ is unbounded above} \\
\Rightarrow &\frac{1}{\|\hat{x}-\hat{y}\|}\left|f(\hat{x},\hat{\omega})g(\hat{y},\hat{\omega}) - f(\hat{y},\hat{\omega})g(\hat{x},\hat{\omega})\right| \text{ is unbounded above since } g(\cdot,\hat{\omega}) \ge \delta_g > 0.
\end{aligned} \tag{12}
$$

We now construct a $L_h(\hat{\omega})$ that bounds (12). We consider four cases. Case I: $f(\hat{x},\hat{\omega}) \ge f(\hat{y},\hat{\omega})$, $g(\hat{x},\hat{\omega}) \ge g(\hat{y},\hat{\omega})$; Case II: $f(\hat{x},\hat{\omega}) \ge f(\hat{y},\hat{\omega})$, $g(\hat{x},\hat{\omega}) \le g(\hat{y},\hat{\omega})$; Case III: $f(\hat{x},\hat{\omega}) \le f(\hat{y},\hat{\omega})$, $g(\hat{x},\hat{\omega}) \ge g(\hat{y},\hat{\omega})$; and Case IV: $f(\hat{x},\hat{\omega}) \le f(\hat{y},\hat{\omega})$, $g(\hat{x},\hat{\omega}) \le g(\hat{y},\hat{\omega})$. In Case I we write $f(\hat{x},\hat{\omega}) = f(\hat{y},\hat{\omega}) + L_f(\hat{x},\hat{\omega})\|\hat{x}-\hat{y}\|$ and $g(\hat{x},\hat{\omega}) = g(\hat{y},\hat{\omega}) + L_g(\hat{x},\hat{\omega})\|\hat{x}-\hat{y}\|$, where $L_f(\hat{x},\hat{\omega}) \le L_f(\hat{\omega})$, $L_g(\hat{x},\hat{\omega}) \le L_g(\hat{\omega})$ are the smallest value of Lipschitz constants for which these equalities hold. Therefore, upon substitution, we have

$$
\frac{1}{\|\hat{x}-\hat{y}\|}\left|f(\hat{x},\hat{\omega})g(\hat{y},\hat{\omega}) - f(\hat{y},\hat{\omega})g(\hat{x},\hat{\omega})\right| = |L_f(\hat{x},\hat{\omega})g(\hat{y},\hat{\omega}) - L_g(\hat{x},\hat{\omega})f(\hat{y},\hat{\omega})| \le
$$

$$
L_f(\hat{\omega})|g(\hat{y},\hat{\omega})| + L_g(\hat{\omega})|f(\hat{y},\hat{\omega})| \le \bar{M}(L_f(\hat{\omega}) + L_g(\hat{\omega})), \bar{M} = \max\{f(\hat{y},\hat{\omega}), g(\hat{y},\hat{\omega})\}
$$

where $\bar{M}$ is a constant due to the boundedness assumption (Assumption A8). In Case II we use $f(\hat{x},\hat{\omega}) = f(\hat{y},\hat{\omega}) + L_f(\hat{x},\hat{\omega})\|\hat{x}-\hat{y}\|$ and $g(\hat{x},\hat{\omega}) = g(\hat{y},\hat{\omega}) - L_g(\hat{y},\hat{\omega})\|\hat{x}-\hat{y}\|$, where $L_g(\hat{y},\hat{\omega})$ is the smallest value of the Lipschitz constant for which the second equality holds. Therefore,

$$
\frac{1}{\|\hat{x}-\hat{y}\|}\left|f(\hat{x},\hat{\omega})g(\hat{y},\hat{\omega}) - f(\hat{y},\hat{\omega})g(\hat{x},\hat{\omega})\right| = |L_f(\hat{x},\hat{\omega})g(\hat{y},\hat{\omega}) + L_g(\hat{y},\hat{\omega})f(\hat{y},\hat{\omega})| \le
$$

$$
L_f(\hat{\omega})|g(\hat{y},\hat{\omega})| + L_g(\hat{\omega})|f(\hat{y},\hat{\omega})| \le \bar{M}(L_f(\hat{\omega}) + L_g(\hat{\omega})), \bar{M} = \max\{f(\hat{y},\hat{\omega}), g(\hat{y},\hat{\omega})\}
$$

Cases III and IV are similar, and consequently, we have

$$
\frac{1}{\|\hat{x}-\hat{y}\|}\left|f(\hat{x},\hat{\omega})g(\hat{y},\hat{\omega}) - f(\hat{y},\hat{\omega})g(\hat{x},\hat{\omega})\right| \le \bar{M}(L_f(\hat{\omega}) + L_g(\hat{\omega})).
$$

By letting, $L_h(\hat{\omega}) = \bar{M}(L_f(\hat{\omega}) + L_g(\hat{\omega}))$, we have a contradiction. Now since

$L_f(\omega) \le M_f, L_g(\omega) \le M_g$ (Assumption A8), $L_h(\omega) \le \bar{M}(M_f + M_g)$. Therefore, $\mathbb{E}[L_h(\omega)^2] \le \bar{M}^2(M_f + M_g)^2$. $\square$

Our next result (Theorem 3.3) states the convergence of $\hat{\phi}_K$ to ϕ^* in distribution under some regularity conditions. It is a direct consequence of the Theorems 3.1 and 3.2 from Shapiro (1991) and Lemma 3.1.

Theorem 3.3. *Let $\mathcal{X}^*(\bar{h})$ be the set of minimizer of $\bar{h}(x)$ over $\mathcal{X}$ and $Z(x)$ is a random element in Banach space for $x \in \mathcal{X}^*(\bar{h})$. Then under Assumptions (A7)-(A8), $\sqrt{K}\,(\hat{\phi}_K - \phi^*)$ converges in distribution to $\min_{x \in \mathcal{X}^*(\bar{h})} Z(x)$. In particular, if $\mathcal{X}^*(\bar{h}) = \{x^*\}$, i.e., the minimizer is unique, then $\sqrt{K}\,(\hat{\phi}_K - \phi^*) \xrightarrow{D} \mathcal{N}(0, \sigma)$ with $\sigma^2 = \mathbb{E}\left[h(x^*, w)^2\right] - \left(\mathbb{E}\left[h(x^*, w)\right]\right)^2$.*

Proof. Recall that $\mathcal{L}_2$ space is an example of Banach space $\mathcal{B}(\mathcal{X})$. Under Assumptions A7 and A8, the sufficient conditions of Theorems 3.1 and 3.2 are satisfied due to Lemma 3.1. $\square$

Next we provide a convergence result for the reformulated problem (10). Similar result can also be established for (11).

Theorem 3.4. *Let Assumptions (A1)-(A8) hold and problem* (10) *has optimal value $\hat{\phi}^r_K$ for some finite K. Assume that x^* is the unique minimizer to the problem $\min_{x \in \mathcal{X}} \mathbb{E}[\frac{f(x,\omega)}{g(x,\omega)}]$. Then $\sqrt{K}\,(\hat{\phi}^r_K - \phi^*) \xrightarrow{D} \mathcal{N}(0, \sigma)$ with $\sigma^2 = \mathbb{E}\left[h(x^*, w)^2\right] - \left(\mathbb{E}\left[h(x^*, w)\right]\right)^2$.*

Proof. Let $\zeta_K = \sqrt{K}\,(\hat{\phi}_K - \phi^*)$ and $\zeta^r_K = \sqrt{K}\,(\hat{\phi}^r_K - \phi^*)$. Then from Theorem 3.3, $\zeta_K \xrightarrow{D} \mathcal{N}(0, \sigma)$ with $\sigma^2 = \mathbb{E}\left[h(x^*, w)^2\right] - \left(\mathbb{E}\left[h(x^*, w)\right]\right)^2$. Let $\Phi_\sigma(\cdot)$ be the CDF of $\mathcal{N}(0, \sigma)$. For any $\epsilon > 0$,

$$\Phi_\sigma(a - \epsilon) \le \lim_{K \to \infty} Pr(\zeta_K \le a) \le \Phi_\sigma(a + \epsilon) \tag{13}$$

For any finite K, optimal value $\hat{\phi}^r_K = \hat{\phi}_K$ (see Proposition (3.1)). Hence,

$$\Phi_\sigma(a - \epsilon) \le \lim_{K \to \infty} Pr(\zeta^r_K \le a) \le \Phi_\sigma(a + \epsilon) \tag{14}$$

Since $\Phi_\sigma(a)$ is continuous at every a, both $\Phi_\sigma(a-\epsilon), \Phi_\sigma(a+\epsilon)$ converge to $\Phi_\sigma(a)$ as $\epsilon \to 0^+$. $\square$

3.3. Reformulation of General Convex-Concave Fractional Program Framework

In formulation (9), we have $d_k > 0$ (see Proof of Proposition 3.1 in Appendix A for details). Multiplying d_k to $c_k/d_k \leq \gamma_k$, we obtain $c_k \leq d_k\gamma_k$ for all $k \in [K]$. Let

$$w_k := \frac{\gamma_k + d_k}{2}, \qquad v_k := \frac{\gamma_k - d_k}{2}. \tag{15}$$

Using $d_k\gamma_k = w_k^2 - v_k^2$, we represent the constraints as $c_k + v_k^2 \leq w_k^2$ for $k \in [K]$. This is a non-convex constraint due to the square term on the right-hand side of the inequality. Using a convex set

$$\mathcal{S} = \left\{ (x, c, d, \theta, \gamma, \pi, w, v) \;\middle|\; \begin{array}{l} f_k(x) \leq c_k \leq \alpha_k, \sigma\beta_k \leq \sigma d_k \leq g_k(x), k \in [K], \\ \gamma_k + d_k = 2w_k, \gamma_k - d_k = 2v_k, k \in [K], \\ f^T\pi \leq \theta, H^T\pi \geq \gamma, P\gamma \leq \theta\mathbb{1}_J \\ x \in \mathcal{X},\ c \in \mathbb{R}^K,\ d \in \mathbb{R}^K,\ \theta \in \mathbb{R}, \\ \gamma \in \mathbb{R}^K,\ \pi \in \mathbb{R}^L, w \in \mathbb{R}^K,\ v \in \mathbb{R}^K. \end{array} \right\}, \tag{16}$$

we obtain an alternative optimization problem of the form

$$\begin{aligned} \vartheta^* := \quad & \min \quad \theta \\ & \text{s.t.} \quad c_k + v_k^2 \leq w_k^2,\ k \in [K],\ (x, c, d, \theta, \gamma, \pi, w, v) \in \mathcal{S}. \end{aligned} \tag{17}$$

Proposition 3.1. *Two optimization problems* (9) *and* (17) *are equivalent:*

- *If* $(x^*, c^*, d^*, \theta^*, \gamma^*, \pi^*)$ *is an optimal solution to* (9)*, then the solution* $(x^*, \hat{c}, \hat{d}, \theta^*, \hat{\gamma}, \pi^*, \hat{w}, \hat{v})$ *such that*

  $$\hat{c}_k = f_k(x^*),\ \hat{d}_k = \sigma g_k(x^*),\ \hat{\gamma}_k = \frac{\hat{c}_k}{\hat{d}_k},\ \hat{w}_k = \frac{\hat{\gamma}_k + \hat{d}_k}{2},\ \hat{v}_k = \frac{\hat{\gamma}_k - \hat{d}_k}{2},$$

  for all $k \in [K]$ *is an optimal solution to* (17)*.*

- *If* $(x^*, c^*, d^*, \theta^*, \gamma^*, \pi^*, w^*, v^*)$ *is an optimal solution to* (17)*, the solution* $(x^*, c^*, d^*, \theta^*, \gamma^*, \pi^*)$ *is an optimal solution to* (9)*.*

Proof. See Appendix A $\square$

3.3.1. Piecewise-Linear Approximations

The reformulated problem (17) has K non-convex constraints of the form $c_k + v_k^2 \leq w_k^2$ for $k \in [K]$. To relax these non-convex constraints, we consider piecewise-linear approximation of w_k^2. Let $W_k := (\mathrm{w}_k^1, \cdots, \mathrm{w}_k^{N_k})$ be a set of points such that $\mathrm{w}_k^1 \leq \cdots \leq \mathrm{w}^{N_k}$ and $1 \leq j < N_k$. The slope of a line passing through two points $\big(w_k^j, (w_k^j)^2\big)$ and $\big(w_k^{j+1}, (w_k^{j+1})^2\big)$ on the (w_k, w_k^2) curve is $\big((w_k^j)^2 - (w_k^{j+1})^2\big)/(w_k^j - w_k^{j+1}) = (w_k^j + w_k^{j+1})$. The intercept of the line with the vertical axis is $(w_k^j)^2 - (w_k^j + w_k^{j+1})w_k^j = w_k^j w_k^{j+1}$. Therefore, the equation of the line passing through consecutive points of W_k is $u(w_k; W_k) = (\mathrm{w}_k^j + \mathrm{w}_k^{j+1})w_k - \mathrm{w}_k^j \mathrm{w}_k^{j+1}$, $1 \leq j < N_k$. Thus we define a piecewise-linear function as

$$u(w_k; W_k) := \max_{1 \leq j < N_k} (\mathrm{w}_k^j + \mathrm{w}_k^{j+1})w_k - \mathrm{w}_k^j \mathrm{w}_k^{j+1}. \tag{18}$$

Proposition 3.2. *For $w_k \in [\mathrm{w}_k^1, \mathrm{w}_k^{N_k}]$, let j be an index such that $\mathrm{w}_k^j \leq w_k \leq \mathrm{w}_k^{j+1}$. Then, we have $w_k^2 \leq (\mathrm{w}_k^j + \mathrm{w}_k^{j+1})w_k - \mathrm{w}_k^j \mathrm{w}_k^{j+1} \leq u(w_k; W_k)$. Moreover, we have*

$$(\mathrm{w}_k^j + \mathrm{w}_k^{j+1})w_k - \mathrm{w}_k^j \mathrm{w}_k^{j+1} - w_k^2 \leq \left(\frac{\mathrm{w}_k^j - \mathrm{w}_k^{j+1}}{2} \right)^2. \tag{19}$$

Proof. We obtain the desired result by observing that $(\mathrm{w}_k^j + \mathrm{w}_k^{j+1})w_k - \mathrm{w}_k^j \mathrm{w}_k^{j+1} - w_k^2 = -(w_k - \mathrm{w}_k^j + \mathrm{w}_k^{j+1}/2)^2 + (\mathrm{w}_k^j - \mathrm{w}_k^{j+1}/2)^2 \geq 0$ for any $w_k \in [\mathrm{w}_j, \mathrm{w}_{j+1}]$. □

Using the piecewise-linear function (18) to approximate the square term w_k^2 in (17), we obtain an approximation problem of the form

$$\begin{aligned}
\bar{\vartheta}(W_1, \cdots, W_k) \ := \ \min \quad & \theta \\
\text{s.t.} \quad & c_k + v_k^2 \leq u(w_k; W_k),\ \mathrm{w}_k^1 \leq w_k \leq \mathrm{w}_k^{N_k},\ k \in [K], \\
& (x, c, d, \theta, \gamma, \pi, w, v) \in \mathcal{S}.
\end{aligned} \tag{20}$$

The following proposition states that $\bar{\vartheta}(W_1, \cdots, W_k)$ serves as a lower bound of

$\vartheta^*(W_1, \cdots, W_k)$ defined as

$$\begin{aligned} \vartheta^*(W_1, \cdots, W_k) \; := \; \min \quad & \theta \\ \text{s.t.} \quad & c_k + v_k^2 \leq w_k^2, \, \mathrm{w}_k^1 \leq w_k \leq \mathrm{w}_k^{N_k}, \, k \in [K], \\ & (x, c, d, \theta, \gamma, \pi, w, v) \in \mathcal{S}. \end{aligned} \tag{21}$$

Proposition 3.3. *Every feasible solution to* (21) *is also feasible to* (20)*. Therefore,* $\bar{\vartheta}(W_1, \cdots, W_k) \leq \vartheta^*(W_1, \cdots, W_k)$.

Proof. It follows using Proposition 3.2. □

In order to model the piecewise linear function $u(w_k; W_k)$, we can use binary variables with SOS2 constraints. However, in our experience solving this mixed binary convex approximation problem is computationally costly especially when N_k is large. Instead of solving this mixed binary convex program, we develop a spatial branch-and-bound algorithm which adaptively refines piecewise linear approximations by dividing the space of $(w_1, w_2, \cdots, w_K)$ into small hyper-rectangles and solves convex approximation problems for sub-hyper-rectangles.

4. An Adaptive Branch-and-Bound Algorithm

Using the idea of piecewise-linear approximations, we introduce a spatial branch-and-bound algorithm to obtain an ϵ-optimal solution to (17). Starting with an initial hyper-rectangle, the algorithm successively breaks it into smaller hyper-rectangles and solves a convex approximation problem for each sub-hyper-rectangle to update the lower bound and the incumbent solution. The algorithm repeats this until the optimality gap becomes smaller than a tolerance level ϵ.

4.1. Initial Hyper-Rectangle

To construct an initial hyper-rectangle, we consider lower and upper bounds of w_k. Let γ_k^m and γ_k^M be lower and upper bounds of $f_k(x)/g_k(x)$ subject to $x \in \mathcal{X}$ for each $k \in [K]$. Since $g_k(x) \neq 0$ for all $x \in \mathcal{X}$ and $\mathcal{X}$ is a compact set, such bounds are well-defined for every $k \in [K]$. Using the definition of w_k in (15), we compute w_k^m and w_k^M using the bounds of γ_k and d_k as $\mathrm{w}_k^m = \gamma_k^m + d_k^m$ and

$\mathrm{w}_k^M = \gamma_k^M + d_k^M$ where d_k^m and d_k^M are lower and upper bounds of $|g_k(x)|$ subject to $x \in \mathcal{X}$ for $k \in [K]$. Using the bounds on w_k, we construct the initial hyper-rectangle $B_0 := [\mathrm{w}_1^m, \mathrm{w}_1^M] \times [\mathrm{w}_2^m, \mathrm{w}_2^M] \times \cdots \times [\mathrm{w}_K^m, \mathrm{w}_K^M]$. If g_k is linear, we can compute tight bounds of d_k^m and d_k^M by solving linear programming problems. For some applications such as equitable resource allocation and Cobb-Doglous production efficiency problems, we are also able to compute tight bounds of γ_k^m or γ_k^M using the Charnes-Cooper transformation (Charnes & Cooper, 1962) as illustrated in Section 7.

4.2. Approximation Problem

Let $B := [\mathrm{w}_1^a, \mathrm{w}_1^b] \times [\mathrm{w}_2^a, \mathrm{w}_2^b] \times \cdots \times [\mathrm{w}_K^a, \mathrm{w}_K^b]$ be a hyper-rectangle such that $B \subset B_0$. For each $k \in [K]$, we use the line passing through $(\mathrm{w}_k^a, (\mathrm{w}_k^a)^2)$ and $(\mathrm{w}_k^b, (\mathrm{w}_k^b)^2)$ to approximate w_k^2 in the interval of $[\mathrm{w}_k^a, \mathrm{w}_k^b]$. By Proposition 3.2, for $w_k \in [\mathrm{w}_k^a, \mathrm{w}_k^b]$, we have $w_k^2 \leq (\mathrm{w}_k^a + \mathrm{w}_k^b) w_k - \mathrm{w}_k^a \mathrm{w}_k^b$. Using this inequality, we obtain a convex approximation problem of the form

$$\begin{aligned} \bar{\vartheta}(B) := \min \quad & \theta \\ \text{s.t.} \quad & c_k + v_k^2 \leq (\mathrm{w}_k^a + \mathrm{w}_k^b) w_k - \mathrm{w}_k^a \mathrm{w}_k^b,\ \mathrm{w}_k^a \leq w_k \leq \mathrm{w}_k^b,\ k \in [K], \\ & (x, c, d, \theta, \gamma, \pi, w, v) \in \mathcal{S}. \end{aligned} \tag{22}$$

Let $\vartheta^*(B)$ be the optimal objective value of (21) with additional box constraints $\mathrm{w}_k^a \leq w_k \leq \mathrm{w}_k^b$ for $k \in [K]$. Then we have $\bar{\vartheta}(B) \leq \vartheta^*(B)$ by Proposition 3.3.

4.3. Evaluation Problem

Let $(\bar{x}(B), \bar{c}(B), \bar{d}(B), \bar{\theta}(B), \bar{\gamma}(B), \bar{\pi}(B), \bar{w}(B), \bar{v}(B))$ be an optimal solution to approximation problem (22). Since $\bar{\theta}(B)$ serves as a lower bound of $\vartheta^*(B)$ for all $B \subset B_0$, taking the minimum of $\bar{\theta}(B)$ for all sub-hyper-rectangles B that partition B_0, we are able to compute a lower bound of ϑ^* in (17). In order to compute an upper bound of ϑ^*, we solve a linear programming problem, which returns the best objective value attainable at $\bar{x}(B)$ as

$$\begin{aligned} \psi(\bar{x}(B)) := \min \quad & \theta \\ \text{s.t.} \quad & \frac{f_k(\bar{x}(B))}{g_k(\bar{x}(B))} \leq \gamma_k, \ k \in [K], \\ & f^T\pi \leq \theta, H^T\pi \geq \gamma, P\gamma \leq \theta \mathbb{1}_J, \theta \in \mathbb{R}, \gamma \in \mathbb{R}^K, \pi \in \mathbb{R}^L. \end{aligned} \tag{23}$$

For any $x \in \mathcal{X}$, $\psi(x)$ serves as an upper bound of ϑ^* since the solution of $\psi(x)$, $(\theta(x), \gamma(x), \pi(x))$, forms a feasible solution to (9) with $(x, c(x), d(x))$ where $c_k = f_k(x), d_k = f_k(x)$ for all $k \in [K]$. Therefore, we compute $\psi(\bar{x}(B))$ each time we obtain $\bar{x}(B)$ and update the incumbent solution if needed.

4.4. Main Algorithm

After constructing the initial hyper-rectangle B_0, we solve the convex approximation problem (22) with $B = B_0$ to obtain $(\bar{x}(B_0), \bar{\vartheta}(B_0))$ and compute $\psi(\bar{x}(B_0))$ by solving the evaluation problem (23). Then, we initialize the incumbent solution, the iteration counter, and the branch-and-bound tree as $(x^0_{\text{CB}}, \theta^0_{\text{CB}}) \leftarrow (\bar{x}(B_0), \psi(\bar{x}(B_0)))$, $t \leftarrow 0$, and $T_0 \leftarrow \{B_0, \bar{\vartheta}(B_0)\}$.

Algorithm 1 SOC-B

optimality tolerance: $\epsilon > 0$
compute bounds on w_k, d_k and construct an initial hyper-rectangle B_0
solve (22) with $B = B_0$ and obtain $(\bar{x}(B_0), \bar{\vartheta}(B_0))$
compute $\psi(\bar{x}(B_0))$ by (23) and let $(x^0_{\text{CB}}, \theta^0_{\text{CB}}) \leftarrow (\bar{x}(B_0), \psi(\bar{x}(B_0)))$
let $\vartheta^t_{\text{CB}} \leftarrow \theta^0_{\text{CB}}$, $t \leftarrow 0$, $T_0 \leftarrow \{(B_0, \bar{\vartheta}(B_0))\}$
while true **do**
  find B_t such that $\bar{\vartheta}(B_t) = \min_{(B,\bar{\vartheta}(B)) \in T_t} \vartheta(B)$ and let $\bar{\vartheta}^t \leftarrow \bar{\vartheta}(B_t)$
  if $(\vartheta^t_{\text{CB}} - \bar{\vartheta}^t)/|\vartheta^t_{\text{CB}}| < \epsilon$ **then**
    return x^t_{CB} and θ^t_{CB}
  else
    let $(x^{t+1}_{\text{CB}}, \theta^{t+1}_{\text{CB}}) \leftarrow (x^t_{\text{CB}}, \theta^t_{\text{CB}})$
    find $k_t = \arg\max_{k \in [K]} (\mathrm{w}^{b,t}_k - \mathrm{w}^{a,t}_k)^2 / d^m_k$ and let B'_t, B''_t as (24), (25)
    for $B \in \{B'_t, B''_t\}$ **do**
      solve (22) with B to obtain $(\bar{x}(B), \bar{\vartheta}(B))$ and (23) for $\psi(\bar{x}(B))$
      if $\psi(\bar{x}(B)) < \theta^{t+1}_{\text{CB}}$ **then**
        update $(x^{t+1}_{\text{CB}}, \theta^{t+1}_{\text{CB}}) \leftarrow (\bar{x}(B), \psi(\bar{x}(B)))$; $\vartheta^{t+1}_{\text{CB}} \leftarrow \theta^{t+1}_{\text{CB}}$
      end if
    end for
    $T_{t+1} \leftarrow T_t \setminus \{(B_t, \bar{\vartheta}(B_t))\} \cup \{(B'_t, \bar{\vartheta}(B'_t))\} \cup \{(B''_t, \bar{\vartheta}(B''_t))\}$
  end if
  $t \leftarrow t + 1$
end while

At each iteration t, we let $B_t := [\mathrm{w}^{a,t}_1, \mathrm{w}^{b,t}_1] \times [\mathrm{w}^{a,t}_2, \mathrm{w}^{b,t}_2] \times \cdots \times [\mathrm{w}^{a,t}_K, \mathrm{w}^{b,t}_K]$ such that $\bar{\vartheta}(B_t) = \min_{(B,\bar{\vartheta}(B)) \in T_t} \vartheta(\bar{B})$ and $\bar{\vartheta}^t \leftarrow \bar{\vartheta}(B_t)$. Note that $\bar{\vartheta}^t$ is the best lower bound until time t since our optimization problem is a minimization problem. If the relative optimality gap, $(\vartheta^t_{\text{CB}} - \bar{\vartheta}^t)/|\vartheta^t_{\text{CB}}|$, is smaller than a tolerance level ϵ, we terminate with an ϵ-optimal solution $(x^t_{\text{CB}}, \theta^t_{\text{CB}})$. Otherwise, let $k_t = \arg\max_k (\mathrm{w}_k^{b,t} - \mathrm{w}_k^{a,t})^2/d_k^m$ and split B_t into B_t' and B_t'' as

$$B_t' := [\mathrm{w}_1^{a,t}, \mathrm{w}_1^{b,t}] \times \cdots \times [(\mathrm{w}_{k_t}^{a,t} + \mathrm{w}_{k_t}^{b,t})/2, \mathrm{w}_{k_t}^{b,t}] \times \cdots \times [\mathrm{w}_K^{a,t}, \mathrm{w}_K^{b,t}], \tag{24}$$

$$B_t'' := [\mathrm{w}_1^{a,t}, \mathrm{w}_1^{b,t}] \times \cdots \times [\mathrm{w}_{k_t}^{a,t}, (\mathrm{w}_{k_t}^{a,t} + \mathrm{w}_{k_t}^{b,t})/2] \times \cdots \times [\mathrm{w}_K^{a,t}, \mathrm{w}_K^{b,t}]. \tag{25}$$

For $B \in \{B_t', B_t''\}$, we solve the convex approximation problem (22) with B to obtain $(\bar{x}(B), \bar{\vartheta}(B))$ and compute $\psi(\bar{x}(B))$ by solving the evaluation problem (23). Comparing $\psi(\bar{x}(B))$ with the current best upper bound $\vartheta_{\mathrm{CB}}^t$, we update the incumbent solution if needed.

Lastly, we update the branch-and-bound tree as

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

The above procedure is repeated until the relative optimality gap becomes smaller than ϵ. A complete summary of the proposed method is given in Algorithm 1.

5. Convergence Analysis

In this section, we provide a convergence analysis for Algorithm 1. Specifically, we provide a bound on the optimality gap $\vartheta_{\mathrm{CB}}^t - \vartheta^* = \theta_{\mathrm{CB}}^t - \theta^* < \epsilon$ as a function of approximation errors at $(\bar{x}^t, \bar{c}^t, \bar{d}^t, \bar{\theta}^t, \bar{\gamma}^t, \bar{\pi}^t, \bar{w}^t, \bar{v}^t)$, which is an optimal solution to (22) with $B = B_t$. Using this bound, we prove the finite convergence of the algorithm. Furthermore, we derive a worst-case bound for the number of iterations to obtain an ϵ-optimal solution.

Since $\mathcal{X}$ is a non-empty compact set, (17) has an finite optimum. Let $(x^*, c^*, d^*, \theta^*, \gamma^*, \pi^*, w^*, v^*)$ be an optimal solution to (17) and M be the max of $\|P\|_\infty := \max_i \sum_j |P_{ij}|$ and $\max\{\|p\|_1 \,|\, p \in \mathcal{P}\}$. We present the bound on the optimality gap in the first part of the following theorem and the worst-case bound of iteration in its second part.

Theorem 5.1. *(a) Let $(\bar{x}^t, \bar{c}^t, \bar{d}^t, \bar{\theta}^t, \bar{\gamma}^t, \bar{\pi}^t, \bar{w}^t, \bar{v}^t)$ and $\bar{\vartheta}^t$ be an optimal solution and the objective value to (22) with $B = B_t$. Then, we have*

$$\vartheta_{CB}^t - \vartheta^* \leq \vartheta_{CB}^t - \bar{\vartheta}^t \leq M \max_{k \in [K]} \frac{\Delta_k}{d_k^m}, \tag{26}$$

where $\Delta_k := (\mathrm{w}_k^{a,t} + \mathrm{w}_k^{b,t})\bar{w}_k^t - \mathrm{w}_k^{a,t}\mathrm{w}_k^{b,t} - (\bar{w}_k^t)^2, \quad k \in [K]. \qquad (27)$

(b) For any $\epsilon > 0$, *let* $n = \sum_{k=1}^{K} n_k$ *where*

$$n_k = \left\lceil \log_2 \sqrt{\frac{M(\mathrm{w}_k^M - \mathrm{w}_k^m)^2}{4\epsilon d_k^m}} \right\rceil, \quad k \in [K]. \tag{28}$$

*Algorithm 1 (*SOC-B*) terminates within* 2^n *iterations.*

Proof. Observe that in (9), $c_k,\ d_k$ are linear terms. Hence, convex-concave/concave-convex fractional program involves linear fractional constraints $c_k/d_k \leq \gamma_k$ and convex constraints from numerator and denominator. All subsequent piecewise linear approximation reformulations are based on these linear fractional constraints only and do not affect convex constraints. Hence this theorem can be proved following the steps in the proof of Theorem 5.1 in Kim & Mehrotra (2021). Specifically when we construct a feasible solution $(\hat{x}^t, \hat{c}^t, \hat{d}^t, \hat{\theta}^t, \hat{\gamma}^t, \hat{\pi}^t, \hat{w}^t, \hat{v}^t)$ to (17) from $(\bar{x}^t, \bar{c}^t, \bar{d}^t, \bar{\theta}^t, \bar{\gamma}^t, \bar{\pi}^t, \bar{w}^t, \bar{v}^t)$ as

$$\begin{aligned}
&\hat{x}^t = \bar{x}^t,\ \hat{c}_k^t = f_k(\bar{x}^t),\ \hat{d}_k^t = g_k(\bar{x}^t),\ \hat{\gamma}_k^t = \frac{f_k(\bar{x}^t)}{g_k(\bar{x}^t)},\\
&\hat{w}_k^t = \frac{1}{2}\left(\frac{f_k(\bar{x}^t)}{g_k(\bar{x}^t)} + g_k(\bar{x}^t)\right),\ \hat{v}_k^t = \frac{1}{2}\left(\frac{f_k(\bar{x}^t)}{g_k(\bar{x}^t)} - g_k(\bar{x}^t)\right) \text{ for } k \in [K] \qquad (29)\\
&\hat{\theta}^t = \bar{\theta}^t + M \max_{k\in[K]} \frac{\Delta_k}{\bar{d}_k^t}, \quad \hat{\pi}^t \in \arg\min f^T\pi \text{ subject to } H^T\pi \geq \hat{\gamma}^t,
\end{aligned}$$

those proof steps can be used since $\hat{\gamma}_k^t \leq f_k(\bar{x}^t)/g_k(\bar{x}^t) \leq \bar{c}_k^t/\bar{d}_k^t$.

Proof of part (b) also follows from the steps in Kim & Mehrotra (2021) (Theorem 5.2) that uses part (a) and the pigeonhole principle. □

Note that 2^{n_k} is in the order of $\mathcal{O}(1/\sqrt{\epsilon})$. This demonstrates the efficiency of SOC-B in achieving solution accuracy.

6. Distributionally Robust Optimization

In this section, we introduce two solution approaches to solve a distributionally robust convex or concave fractional program with finite support.

6.1. Dual Reformulation

We first consider the case where the ambiguity set is polyhedral. Let $\mathcal{P}$ be a polyhedral ambiguity set as defined in (A4). Then, the distributionally robust convex or concave fractional program with finite support is formulated as

$$\begin{aligned} \min \quad & \theta \\ \text{s.t.} \quad & f_k(x) \leq c_k \leq \alpha_k,\ \sigma\beta_k \leq \sigma d_k \leq g_k(x),\ \frac{c_k}{d_k} \leq \gamma_k,\ k \in [K], \\ & x \in \mathcal{X}, p^T\gamma \leq \theta, \forall \{p \in \mathbb{R}^K | Hp = f, p \geq 0\}, \theta \in \mathbb{R}, c \in \mathbb{R}^K, d \in \mathbb{R}^K, \gamma \in \mathbb{R}^K \end{aligned} \tag{30}$$

Using the linear programming duality, we can reformulate (30) as follows.

Proposition 6.1. *Optimization problem* (30) *is equivalent to*

$$\begin{aligned} \min \quad & \theta \\ \text{s.t.} \quad & f_k(x) \leq c_k \leq \alpha_k,\ \sigma\beta_k \leq \sigma d_k \leq g_k(x),\ \frac{c_k}{d_k} \leq \gamma_k,\ k \in [K], \\ & f^T\pi \leq \theta, H^T\pi \geq \gamma, x \in \mathcal{X}, \theta \in \mathbb{R}, c \in \mathbb{R}^K, d \in \mathbb{R}^K, \gamma \in \mathbb{R}^K, \pi \in \mathbb{R}^L \end{aligned} \tag{31}$$

Proof. This follows from the linear programming duality. □

Since the above reformulated problem (31) is an instance of (9), we can use SOC-B to solve it. Note that many finitely supported ambiguity sets are polyhedral. For the dualized reformulations with polyhedral ambiguity sets, see Kim & Mehrotra (2021) and Luo & Mehrotra (2020).

6.2. Cutting Surface Algorithm

Next, we introduce an iterative approach to solve a distributionally robust fractional program in the form

$$\begin{aligned} \min \quad & \theta \\ \text{s.t.} \quad & f_k(x) \leq c_k \leq \alpha_k,\ \sigma\beta_k \leq \sigma d_k \leq g_k(x),\ \frac{c_k}{d_k} \leq \gamma_k,\ k \in [K], \\ & p^T\gamma \leq \theta,\ \forall\, p \in \mathcal{C},\ x \in \mathcal{X},\ \theta \in \mathbb{R},\ c \in \mathbb{R}^K,\ d \in \mathbb{R}^K,\ \gamma \in \mathbb{R}^K, \end{aligned} \tag{32}$$

where $\mathcal{C}$ is any convex ambiguity set. Problem (32) is a semi-infinite program due to the presence of constraints $p^T\gamma \leq \theta,\ \forall\, p \in \mathcal{C}$. In the special case when $\mathcal{C}$ is an ellipsoid we can replace $\left\{ \max_{p\in\mathcal{C}} p^T\gamma \leq \theta \right\}$ with an explicit expression for the optimal value. For more general convex ambiguity sets, many works (Wiesemann et al., 2014; Bertsimas et al., 2010; Delage & Ye, 2010) dualize this problem under the assumptions allowing for strong duality. However, unlike the case with polyhedral ambiguity sets, strong duality does not necessarily hold for convex ambiguity sets if the regularity conditions are not satisfied and it may not be always possible to check the regularity conditions. Thus to develop an algorithm applicable in a general setting, we discuss an alternative approach based on the cutting surface algorithm (Mehrotra & Papp, 2014; Luo & Mehrotra, 2019) below. The cutting surface algorithm assumes that an oracle is available to generate a separating probability cut.

To solve the semi-infinite problem (32), we consider a sequence of problems of the form

$$\begin{aligned} \min \quad & \theta \\ \text{s.t.} \quad & f_k(x) \leq c_k \leq \alpha_k,\ \sigma\beta_k \leq \sigma d_k \leq g_k(x),\ \frac{c_k}{d_k} \leq \gamma_k,\ k \in [K], \\ & P^t\gamma \leq \theta \mathbb{1}_t,\ x \in \mathcal{X},\ \theta \in \mathbb{R},\ c \in \mathbb{R}^K,\ d \in \mathbb{R}^K,\ \gamma \in \mathbb{R}^K, \end{aligned} \tag{33}$$

where each row of P^t is an element of a finite set $\mathcal{C}^t := \{p^0, p^1, \cdots, p^t\} \subset \mathcal{C}$ and p^0 is an empirical distribution. Let $(x^t, \theta^t, c^t, d^t, \gamma^t)$ be an optimal solution to (33) at iteration t and p^{t+1} be an $\epsilon/2$-optimal solution to the separation problem $\max_{p\in\mathcal{C}} p^T\gamma^t$. If $(p^{t+1})^T\gamma^t - \theta^t \leq \epsilon/2$ holds, the algorithm terminates with the solution $(x^t, \theta^t, c^t, d^t, \gamma^t)$. Otherwise, we add a probability cut p^{t+1} to $\mathcal{C}^t$ and repeat the above process. Please see Algorithm 2 for a summary of the cutting surface algorithm.

Algorithm 2 A cutting surface algorithm for (32)

Input: optimality tolerance $\epsilon > 0$, empirical distribution p^0.
Step 1: $\mathcal{C}^0 \leftarrow \{p^0\}$, $t \leftarrow 0$.
Step 2: Find an optimal solution $(x^t, \theta^t, c^t, d^t, \gamma^t)$ of (33) with $\mathcal{C}^t$.
Step 3: Find an $\epsilon/2$-optimal solution p^{t+1} of the problem $\max_{p\in\mathcal{C}} p^T\gamma^t$.
Step 4: If $(p^{t+1})^T\gamma^t - \theta^t \le \epsilon/2$, stop and return $(x^t, \theta^t, c^t, d^t, \gamma^t)$;
otherwise $\mathcal{C}^{t+1} \leftarrow \mathcal{C}^t \cup \{p^{t+1}\}$, $t \leftarrow t+1$, and go to Step 2.

Let $\theta^M := \max_{k\in[K]} \gamma_k^M$ and $\Gamma := \{(x, \theta, c, d, \gamma) \mid f_k(x) \le c_k \le \alpha_k,\ \sigma\beta_k \le \sigma d_k \le g_k(x),\ c_k/d_k \le \gamma_k \le \gamma_k^M,\ p^T\gamma \le \theta \le \theta^M,\ x \in \mathcal{X},\ \theta \in \mathbb{R}, c \in \mathbb{R}^K, d \in \mathbb{R}^K, \gamma \in \mathbb{R}^K\}$.

Theorem 6.1. *Suppose that $\mathcal{C}$ is a compact set such that $\sum_{k=1}^K p_k = 1$ and $p \ge 0$ for all $p \in \mathcal{C}$. Then, Algorithm 2 returns an ϵ-optimal solution in a finite number of iterations.*

Proof. By Proposition 3.1, without loss of generality, we can assume that $\gamma^m \le \gamma^t \le \gamma^M$. Since (32) minimizes θ, there exists some $0 \le j \le t$ such that $(p^j)^T\gamma^t = \theta$. From $\sum_{k=1}^K p_k^j = 1$ and $p^j \ge 0$, we have $\theta \le \theta^M$. Therefore, $(x^t, \theta^t, c^t, d^t, \gamma^t) \in \Gamma$ holds for all $t \ge 0$.

Since Γ is closed and bounded, Γ is compact. Also, since $\mathcal{C}$ is compact, so is $\Gamma \times \mathcal{C}$. From that $g(x, \theta, c, d, \gamma) := \gamma^T p - \theta$ is continuous on $\Gamma \times \mathcal{C}$, by (Luo & Mehrotra, 2019, Theorem 3.2), we obtain the desired result. $\square$

7. Computational Performance

The algorithm presented so far is called SOC-B. In Section 7.1, we propose a modified branching strategy for SOC-B that can make SOC-B more efficient. We next discuss implementation details for the algorithms implemented. Subsequently, we discuss the computational results for the two models introduced in Section 1.1.

7.1. Modification of SOC-B

In this section, we present a modification to SOC-B (Algorithm 3 in Appendix B). It has a branching strategy based on an LP relaxation of the convex constraints from the functions such as $f_k(x) \leq c_k, d_k \leq g_k(x)$, $k \in [K]$ arising in the model.

In comparison to Algorithm 1, Algorithm 3 executes several new commands throughout lines 10-17, 24, 26-30. Recall that the former solves the sub-problems for every hyper-rectangles $\{B'_t, B''_t\}$ partitioned from the current active hyper-rectangle B_t. However, the latter does so in lines 31-37 only when both LP relaxation-based conditions in Lines 27 and 29 fail. Under first condition at Line 27, if any objective value from relaxed subproblem corresponding to $B \in \{B'_t, B''_t\}$ is greater than the current globally valid upper bound θ^{t+1}_{CB}, we can fathom that hyper-rectangle. Second condition (Line 29) only works for second hyper-rectangle B''_t (partitioned from B_t) if (i) we already fathomed its companion hyper-rectangle B'_t but failed to fathom it by line 27, and (ii) length of the currently considered edge k_t (chosen as per line 23) is smaller than a threshold.

When conditions at line 29 are satisfied, instead of immediately evaluating the corresponding subproblem, we keep them on hold and record those hyper-rectangles via a set $\tilde{T}$. By doing so we are just changing the priority rule for their evaluation as there is less chance to get $\epsilon-$optimal solution from such a hyper-rectangle B''_t. We may already have achieved $\epsilon-$optimality from some other more competitive hyper-rectangle before revisiting them (via line 11-16). Even when we require to further branch the hyper-rectangles from $\tilde{T}$, some of them become fathomable because of the updated global upper bound (Line 12). We do such priority based ordering only once as indicated by a switching variable $tree2$. In particular, once we start evaluating those sorted subproblems, we do not further sort them. Note that finite convergence criteria remains unaffected due to this modifications as termination happens only when relative optimality gap is below a given tolerance.

7.2. Implementation Details

All the computations are performed using a 64-core server with Xeon 2.20 GHz CPUs and 128 GB RAM. For each problem size, we generate multiple instances. For solving each of these instances, only one core is used. The code is written using Python programming language, particularly, python 3.7. We use the following python packages: numpy, copy, math, queue, time, cyipopt. We also use dictionary and PriorityQueue data structure in our implementation. We use GUROBI and IPOPT to solve optimization problems arising in our implementation.

Package 'numpy' is used for random data generation of the instances. Package 'scikit-learn' is used for parameter normalization in Cobb-Douglas problem. Linear and mixed integer linear bound computation problems in equitable resource allocation problem are solved using GUROBI. On the other hand, the bound providing nonlinear convex problems in Cobb-Douglas problem are solved using IPOPT solver of the cyipopt package. In branch-and-bound type algorithm implementation, data structure 'PriorityQueue' from python package 'queue' is used so that the leaf node information having least objective value can easily be accessed. Additionally, a package called 'time' is used to keep track of computation time. GUROBI is used for solving the linear optimization problems arising in the branch-and-bound tree. IPOPT is used to solve convex optimization problems. Since IPOPT uses a starting solution as an input, the optimal solution to a current subproblem is utilized later as the starting point for its two branch subproblems.

7.3. Equitable Resource Allocation Problem

7.3.1. Data Generation

The instances of the equitable resource allocation problem (4) were created as follows for each scenario $k \in [K]$. For each supplier $i \in [m]$, we generated the amount of available resource $r_i^k \sim \text{Uniform}(0, n)$. For each customer $j \in [n]$, we generated the demand $d_j^k \sim \text{Uniform}(0, m)$. For each $i \in [m]$ and $j \in [n]$, we let $b_{ij}^k \sim \text{Uniform}(0, 1)$ which is the benefit each unit resource from supplier i brings to customer j. For data dimensions, we consider $n = 15, m \in \{5, 10, 15\}$ while number of scenarios $k \in \{5, 10\}$. Thus under each scenario the largest instance has 225 decision variables. We generated five instances for each dimension combination. For the first part of the experiments, we use the allocation threshold parameter $\delta = 0.8$.

7.3.2. Bounds Computation

In (5), we have $f_k(x) = \sum_{j=1}^{n} |\sum_{i=1}^{m} b_{ij}^k x_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij}^k x_{ij}|/n$ and $g_k(x) = |\sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij}^k x_{ij}|$. By data generation, $\sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij}^k x_{ij} \geq 0$ holds for all feasible x for all k. Observe that optimal value to (3) satisfies $\theta^* \geq 0$. If $\theta^* = 0$, then the equity model is not of interest. Hence, without loss of generality we assume that $\theta^* > 0$, and thus Assumption (A5) holds because $\mathcal{X}$ is a non-empty compact set. Moreover, $\sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij} x_{ij} \geq \delta\theta^*$ is valid for all $\delta \in [0, 1]$. Therefore, considering linearity of $g_k(x)$ we can compute d_k^m and d_k^M by solving linear programming problems. Also, we let $\gamma_k^m = 0$ since $f_k(x) \geq 0$ for all feasible x. To compute an upper bound γ_k^M, we use the Charnes-Cooper transformation (Charnes & Cooper, 1962) as $\gamma_k^M = \max \sum_{j=1}^{n} |\sum_{i=1}^{m} b_{ij}^k y_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij}^k y_{ij}|/n$ subject to $\sum_{j=1}^{n} y_{ij} \leq r_i^k t$, $i \in [m]$, $\sum_{i=1}^{m} b_{ij}^k y_{ij} \leq d_j^k t$, $j \in [n]$, $\sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij}^k y_{ij} \geq \delta\theta^{k*} t$, $\sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij}^k y_{ij} = 1$, $t \geq 0$ and $y_{ij} \geq 0$, $i \in [m]$, $j \in [n]$. Since the problem is non-convex, we solve its mixed-binary reformulation using the Big-M technique. This technique is also used for computing $\alpha_k^M = \max_{x \in \mathcal{X}^k} f_k(x)$.

7.3.3. Experimental Results

For each problem size (m, n, k), we run the algorithms with a 12-hour time limit. Multiple (five) instances for each problem size were generated to see the variability in computation time. We report computation times on Table 1 when desired five-digit optimality gap is achieved.

Table 1 shows that SOC-B attains the relative optimality tolerance of $\epsilon = 10^{-5}$ for most of the instances except few $K = 10$ instances. Benson's algorithm does not attain desired five-digit optimality gap for any of $K = 5$ and $K =$

Table 1: Solution time (s) of SOC-B to attain $\epsilon = 10^{-5}$ optimality gap for the stochastic equitable resource allocation problem. No value is reported if the solution time exceeded 12 hours.

Problems Size		SOC-B				
k	m	Inst. = 1	2	3	4	5
5	5	202	281	113	446	410
	10	728	291	125	283	478
	15	806	457	529	628	1249
10	5	6034	7147	13285	11434	6373
	10	25485	31839	12378		
	15	30152	15840			

10 instances within the time limit. In fact, using Benson Algorithm, most $K = 5$ instances attain no digit accuracy was achieved, except in two instances that achieved one digit accuracy. None of $K = 10$ instances achieved one-digit accuracy. Overall, the optimality gap attained by Benson's algorithm ranges between 5.96×10^{-2} and 8.01×10^{-1}. On the other hand, this gap for SOC-B ranges from $3.99 \times 10^{-5} \sim 3.62 \times 10^{-3}$ for the instances where five-digit accuracy was not attained. These results clearly demonstrate the efficiency of SOC-B for solving convex fractional programs.

7.3.4. Equity-Efficiency Analysis

Table 2 reports average objective values for the allocation threshold parameter $\delta \in \{0.6, 0.7, 0.8, 0.9, 1.0\}$. The standard error ranges from 0.016 to 0.115. The case with no equity consideration corresponds to $\delta = 1$ while the case with full equity consideration corresponds to $\delta = 0$, completely ignoring the optimal objective value obtained from the benefit maximization model. As expected, the objective value (unfairness) increases as equity considerations reduce (δ increase). Compared to the worst case ($\delta = 1$), about 66% to 84% improvement in fairness is achieved when ensuring at least 80% of the maximum benefit ($\delta = 0.8$).

Table 2: Sensitivity analysis of the effect of δ on fairness. For each problem instance, we run SOC-B algorithm on five instances and report the average objective values and corresponding standard errors in parentheses.

Problem Size		δ				
m	k	0.6	0.7	0.8	0.9	1.0
5	5	0.18997 (0.04836)	0.21683 (0.05032)	0.27018 (0.05269)	0.46846 (0.08327)	1.10960 (0.08278)
	10	0.19424 (0.05036)	0.22166 (0.05803)	0.26535 (0.06032)	0.39030 (0.06313)	0.98559 (0.09089)
	15	0.07286 (0.01755)	0.09170 (0.01952)	0.11760 (0.02238)	0.2103 (0.02937)	0.77537 (0.04703)
10	5	0.27877 (0.03275)	0.30971 (0.03635)	0.34698 (0.03777)	0.50834 (0.05389)	1.04873 (0.05270)
	10	0.14869 (0.01627)	0.17821 (0.01997)	0.21434 (0.02246)	0.31869 (0.03438)	0.82300 (0.11581)
	15	0.14162 (0.03094)	0.16534 (0.03077)	0.19617 (0.02981)	0.30798 (0.05059)	0.92582 (0.06196)

7.4. Stochastic Cobb-Douglas Production Efficiency Problem

In this section we consider the stochastic and distributionally robust Cobb-Douglas production efficiency problem (8) with finite support. The sample average formulation of the problem is given as follows:

$$\max_{x\in\mathcal{X}} \quad \sum_{k=1}^{K} \frac{1}{K}\left[\frac{\prod_{j=1}^{n} a_{k0}\, x_j^{a_{kj}}}{\sum_{j=1}^{n} c_{kj}x_j + c_{k0}}\right]. \tag{34}$$

Let $\mathcal{X} := \{x \,|\, Ax \leq b, x \geq 0\}$. The distributionally robust variant using the finitely supported Wasserstein ambiguity set is discussed in Section 7.5.

7.4.1. Data Generation

Let A_{ij} be the element in the i^{th} row and the j^{th} column of matrix A. For each $i \in [m]$ and $j \in [n]$, we let $A_{ij} \sim \text{Uniform}(0,1)$ and $b_i = n$ for all $i \in [m]$. On the other hand, for each scenario $k \in [K]$, we generate $c_{k0} \sim \text{Uniform}(1,2)$ and $c_{kj} \sim \text{Uniform}(0,1)$ for each $j \in [n]$. For the Cobb-Douglas functions, we let $a_{k0} \sim \text{Uniform}(1,2)$ and $a_{kj} \sim \text{Uniform}(0,1)$. Then, we divide a_{kj} by $\sum_{j=1}^{n} a_{kj}$ so that $\sum_{j=1}^{n} a_{kj} = 1$ holds for all $k \in [K]$. For data dimensions, we consider $n \in [5,10,20]$, $K \in [5,10]$, and $m = \lceil n/2 \rceil$. For each dimension, we generate five instances to have an insight in the computational performance differences.

7.4.2. Bounds Computation

Putting (34) in the form of (9), we have $f_k(x) = -\prod_{j=1}^{n} a_{k0}\, x_j^{a_{kj}}$ and $g_k(x) = -\sum_{j=1}^{n} c_{kj}x_j + c_{k0}$ with $\sigma = -1$. We compute d_j^m and d_j^M by solving linear programming problems. For γ_k^m and γ_k^M, we compute γ_k^m using the Charnes-Cooper transformation (Charnes & Cooper, 1962) as $\gamma_k^m := -\max \prod_{j=1}^n a_{k0}\, y_j^{a_{kj}}$ subject to $Ay \leq bt$, $y \geq 0$, and $\sum_{j=1}^n c_{kj} y_j + c_{k0} = 1$. We let $\gamma_k^M = 0$, and set $\alpha_k = 0$ and $\beta_k = d_k^M$ in (9).

7.4.3. Computational Experience with Stochastic Cobb-Douglas Model

For our computational comparison, we implemented SOC-B (Algorithm 1) along with two benchmark algorithms (B-G (Benson, 2002b), B-J (Benson, 2002a)). For each problem size (n, K), we run the algorithms on five instances with a 12-hour time limit and an optimality tolerance of 10^{-5}. We report computation times for SOC-B and relative optimality gaps upon termination for the other algorithms in Table 3.

Our experimental results show that B-G and B-J generally fail to achieve five digits of accuracy in 12 hours. However, SOC-B attains this accuracy for all instances of any problem size. Moderate size models such that $(n, K) \in \{(5,5), (5,10), (10,5), (10,10), (15,5)\}$ are mostly solved within two hours. These results clearly demonstrate that SOC-B achieves a significant reduction in computational time in solving stochastic concave fractional programs. However, the difficulty in finding a solution with the desired level of accuracy increases with increase in the problem dimension and the number of scenarios.

7.4.4. Efficient implementation of SOC-B for Stochastic Cobb-Douglas Model

Since numerators in stochastic Cobb-Douglas fractional problem are nonlinear, we used the modified algorithm outlined in Section 7.1 for all the instances stochastic of Cobb Douglas problem and reported them in Table 3 under SOC-B (efficient). Numerical findings show that, in comparison to the original algorithm, the average reduction in computational time is about 47.8%. Maximum improvement is 89.8% while 16 instances out of 30 instances exhibit more than 40% improvement.

Table 3: Experimental results for stochastic Cobb-Douglas production efficiency problem.

Problem		n	5		10		15	
		K	5	10	5	10	5	10
SOC-B (Algorithm 1)	Time (s)	1	29	1073	135	4288	353	30000
		2	30	1599	208	6948	278	24771
		3	17	797	109	9790	199	34693
		4	25	1012	102	4017	299	21181
		5	23	785	125	7546	308	16489
SOC-B (Algorithm 3)	Time (s)	1	23	231	65	2160	186	15732
		2	20	320	151	4553	177	19579
		3	3	195	39	5290	128	11062
		4	11	103	89	1292	205	16684
		5	6	266	90	4929	233	11916
B-G	Opt. Gap (rel)	1	1.21E-04	2.09E-03	2.27E-04	2.60E-03	2.88E-04	5.32E-03
		2	1.26E-04	3.62E-03	3.23E-04	8.37E-03	3.88E-04	6.01E-03
		3	1.96E-05	1.23E-03	1.44E-04	6.17E-03	2.34E-04	2.96E-03
		4	6.02E-05	2.51E-03	1.22E-04	2.24E-03	3.16E-04	8.66E-03
		5	4.15E-05	1.78E-03	2.62E-04	4.62E-03	4.71E-04	6.30E-03
B-J	Opt. Gap (rel)	1	4.10E-04	9.59E-03	7.18E-04	1.45E-02	7.84E-04	1.91E-02
		2	4.83E-04	1.14E-02	9.24E-04	2.09E-02	9.56E-04	2.42E-02
		3	7.53E-05	4.18E-03	4.59E-04	2.24E-02	6.91E-04	1.64E-02
		4	1.69E-04	7.64E-03	3.67E-04	1.46E-02	1.12E-03	2.23E-02
		5	1.40E-04	1.02E-02	7.40E-04	1.94E-02	1.38E-03	2.33E-02

7.5. Distributionally Robust Cobb-Douglas Production Efficiency Problem

We next present experimental results for the proposed solution approach for solving distributionally robust Cobb-Douglas production efficiency problems. In these experiments we use the dual formulation (DUAL) based approach and the cutting-surface algorithm (CUT). Here our interest is also to study the performance of the algorithms with increasing ambiguity.

7.5.1. Data Generation

For each $(n, K) \in \{(5,5),(5,10),(10,5),(10,10),(15,5),(15,10)\}$, we consider three types of underlying distributions to investigate the performance of DUAL and CUT algorithms. For $k, j \in [K]$, we sample a_{k0}, a_{kj}, c_{k0}, c_{kj} according to the following probability distributions:

- Uniform: $a_{k0}, c_{k0} \sim \text{Uniform}(1,2)$, $a_{kj}, c_{kj} \sim \text{Uniform}(0,1)$.
- Left-Skewed: $a_{k0}, c_{k0} \sim 1 + \text{Beta}(5,2)$, $a_{kj}, c_{kj} \sim \text{Beta}(5,2)$.
- Right-Skewed: $a_{k0}, c_{k0} \sim 1 + \text{Beta}(2,5)$, $a_{kj}, c_{kj} \sim \text{Beta}(2,5)$.

After generating $a_{k1}, \cdots, a_{kn}$, we divide a_{kj} by $\sum_{j=1}^{n} a_{kj}$ so that $\sum_{j=1}^{n} a_{kj} = 1$ holds for all $k \in [K]$. On the other hand, we sample A and b according to the procedure in Section 7.4.1 and use the same one for all three instances.

7.5.2. Dual Formulation with Wasserstein Ambiguity Set

Let $\Delta^{\max}$ be the maximum Wasserstein distance from the nominal (empirical) probability distribution p^0 computed as the max of $\sum_{i=1}^{K}\sum_{j=1}^{K} q_{ij} d(\xi_i, \xi_j)$ subject to $\sum_{j=1}^{K} q_{ij} = p_i$, $i \in [K]$, $\sum_{i=1}^{K} q_{ij} = p_j^0$, $j \in [K]$, $\sum_{k=1}^{K} p_k = 1$, $p_k \geq 0$, $k \in [K]$, $q_{ij} \geq 0$, $i, j \in [K]$. $d(\xi_i, \xi_j)$ is the Euclidean distance between two vectors ξ_i and ξ_j. Note that $p_j^0 = 1/K$ for all $j \in [K]$ in (34). We use the Wasserstein radius of $\Delta := \rho\Delta^{\max}$ where $\rho \in \{0.01, 0.05, 0.1\}$. Thus the ambiguity set $\{p \in \mathbb{R}^K | Hp = f,\, p \geq 0\}$ in (30) is given as:

$$\mathcal{D}_W = \left\{ p \in \mathbb{R}_+^K \;\middle|\; \begin{array}{l} \exists\, q \in \mathbb{R}_+^{K\times K} \;: \\ \sum_{j=1}^{K} q_{ij} - p_i = 0\, \forall i \in [K],\, \sum_{i=1}^{K} q_{ij} = p_j^0\, \forall j \in [K] \\ \sum_{k=1}^{K} p_k = 1,\, p_k \geq 0\ \forall k \in [K],\, q_{ij} \geq 0, \forall i, j \in [k] \\ \sum_{i=1}^{K}\sum_{j=1}^{K} q_{ij} d(\xi_i, \xi_j) \leq \rho\Delta^{\max} \end{array} \right\} \tag{35}$$

The corresponding dual formulation is stated as:

$$\begin{aligned} \min \quad & -\sum_{k=1}^{K} p^0 t_k - \Delta\nu + \varsigma \\ \text{s.t.} \quad & -\prod_{j=1}^{n} a_{k0}\, x_j^{a_{kj}} \leq c_k \leq 0,\; \sum_{j=1}^{n} c_{kj} x_j + c_{k0} \leq d_k \leq d_k^M,\; k \in [K], \\ & -s_k - r_k + \varsigma \geq \gamma_k,\; \frac{c_k}{d_k} \leq \gamma_k,\; k \in [K], \\ & -s_i + t_j + d(\xi_i, \xi_j)\nu + \lambda_{ij} \leq 0,\; i, j \in [K], \\ & x \in \mathcal{X},\, s \in \mathbb{R}^K,\, t \in \mathbb{R}^K,\, r \in \mathbb{R}_+^K,\, \varsigma \in \mathbb{R},\, \nu \leq 0,\, \lambda \in \mathbb{R}_+^{K\times K}. \end{aligned} \tag{36}$$

7.5.3. Computational Experience with Distributionally Robust Cobb-Douglas Model

Table 4 summarizes the experimental results for distributionally robust Cobb-Douglas production efficiency problem. For each problem size and probability distribution, we generate a problem instance and run the algorithms for three

Table 4: Experimental results of distributionally robust Cobb-Douglas production model.

Distribution			Uniform				Left-Skewed				Right-Skewed			
Problem			DUAL		CUT		DUAL		CUT		DUAL		CUT	
n	K	ρ	Obj. Val	Time(s)	Time(s)	Cuts	Obj. Val	Time(s)	Time(s)	Cuts	Obj. Val	Time(s)	Time(s)	Cuts
5	5	0.01	0.1902	48	47	1	0.06340	17	79	1	0.02575	6	73	1
		0.05	0.1846	59	50	1	0.06198	84	49	1	0.02487	6	78	1
		0.1	0.1776	51	50	1	0.06023	18	84	1	0.02382	6	77	1
	10	0.01	0.2210	706	971	1	0.08587	271	190	1	0.04292	1976	1849	1
		0.05	0.2109	702	1933	2	0.08307	71	536	1	0.03973	1815	3729	2
		0.1	0.1997	835	4589	2	0.07981	4482	2264	1	0.03576	1966	4316	2
10	5	0.01	0.1148	94	127	1	0.03321	103	317	1	0.01362	2124	2969	1
		0.05	0.1108	119	152	1	0.03241	194	336	1	0.01270	3197	3100	1
		0.1	0.1061	122	153	1	0.03142	168	338	1	0.01166	2217	3193	1
	10	0.01	0.1133	3744	5831	2	0.03419	27383	30377	1	0.02105	199514	160505	1
		0.05	0.1098	3379	6242	2	0.03290	40001	39928	1	0.01970	240094	157936	1
		0.1	0.1054	3442	17239	3	0.03149	79311	88858	1	0.01818	249672	170508	1
15	5	0.01	0.07796	202	321	1	0.03642	337	588	1	0.006737	12092	23769	1
		0.05	0.07653	176	314	1	0.03607	384	606	1	0.006133	19007	28340	1
		0.1	0.07491	157	524	2	0.03565	339	646	1	0.005417	26816	35249	1
	10	0.01	0.07506	26969	30066	1	0.02620	122126	159051	1	0.01569	439119	507505	1
		0.05	0.07269	26388	38480	1	0.02525	147633	136761	1	0.01475	413556	507311	1
		0.1	0.07055	52908	301819	3	0.02439	170839	144876	1	0.01363	550554	561600	1

different values of ρ. We run the dual (DUAL) formulation and cutting surface (CUT) solution approaches until they attain the relative optimality gap of 10^{-5}. For both approaches, we incorporated the efficient strategy proposed in Algorithm 3. We report objective values, computation times and the number of probability cuts needed in the cutting surface algorithm. In these experiments we did not impose a time limit, allowing us to make a more complete comparison.

Computational results suggest that when compared to the performance reported in Table 3 the DUAL algorithm takes two to three times more computation time than the time required to solve the underlying stochastic programs. While the CUT algorithm mostly required only one or two probability cut for our test instances, the DUAL algorithm still tends to be more efficient than the CUT algorithm. We also observe that the computation times of DUAL algorithm remains similar for different values of Δ. However, the time required by the CUT algorithm increases with Δ, as more probability cuts are required in this case. This is because a new non-convex optimization problem is solved after the addition of a probability cut. Even though the new problem is solved with the initial point obtained using the strategy mentioned in Section 7.2, the time required to solve multiple problems with sufficient accuracy is not offset despite it being in a lower dimension. Lastly, computation times vary widely across parameter distributions. It takes less time to solve problem instances from the uniform distribution than those from the skewed distributions. Among the skewed distributions, larger size instances generated from the right-skewed distribution take substantially more computation time than those generated from the left-skewed distribution. The reasons for this phenomenon are unclear.

8. Concluding Remarks

We studied convex and concave fractional programs as well as their stochastic counterparts in a common framework. The proposed branch-and-bound algorithm efficiently finds a highly accuracy solution to moderate size stochastic equitable resource allocation and Cobb-Douglas problem instances. Although the problem difficulty does grow rapidly with number of scenarios and variables in the problem, the algorithm developed in this paper is a significant advancement over previously known algorithms that can be used for solving such problems.

The distributionally robust problems were studied under the finite support assumption. This can be extended to the compact continuous support counterpart. To see this, let us consider a compact continuous support Ξ for random parameter ξ and l_1-Wasserstein distance-based ambiguity set for the unknown distribution $\mathbb{P} \in \mathcal{M}(\Xi)$. Additionally assume $\mathbb{Q}$ is the nominal distribution available on finite support $\Xi^S = \{\xi^1, \xi^2, \cdots, \xi^S\}$, i.e., $\mathbb{Q} \in \mathcal{M}(\Xi^S)$ and for each of the support it takes equal probability $1/S$. Then, corresponding continuously supported distributionally-robust convex-concave fractional problem (CS-DR-CCFP) is given as:

$$\min_{x\in\mathcal{X}} \max_{\mathbb{P}\in\mathcal{D}} \int_{\xi\in\Xi} \frac{f(x,\xi)}{g(x,\xi)}\, d\mathbb{P}(\xi), \tag{CS-DR-CCFP}$$

where

$$\mathcal{D} = \left\{ \mathbb{P} \in \mathcal{M}(\Xi) \;\middle|\; \begin{array}{l} \exists\, \Pi \in \mathcal{J}(\Xi \times \Xi^S) \;: \\ \sum_{s=1}^{S} \int_{\Xi} ||\xi - \xi^s||_1 \Pi(\xi, \xi^s) d\xi \leq \tau \\ \int_{\Xi} \Pi(\xi, \xi^s) d\xi = \frac{1}{S}, \quad s \in [S] \\ \sum_{s \in [S]} \Pi(\xi, \xi^s) = d\mathbb{P}(\xi) \;\; \forall \xi \in \Xi \end{array} \right\}, \tag{Wass}$$

Now if $h(x, \xi) = \frac{f(x,\xi)}{g(x,\xi)}$ and $h(\cdot, \cdot)$ is bounded on compact space $\mathcal{X} \times \Xi$, by Theorem 3.1 of Luo & Mehrotra (2019) strong duality holds for the inner maximization problem. Thus, (CS-DR-CCFP) can equivalently be written as

$$\begin{aligned} \min_{x \in \mathcal{X}, \mu \geq 0} \quad & \tau\mu + \frac{1}{S} \sum_{s \in [S]} \nu^s \\ \text{s.t.} \quad & \max_{\xi \in \Xi} h(x, \xi) - \mu ||\xi - \xi^s||_1 \leq \nu^s \quad \forall s \in [S] \end{aligned} \tag{37}$$

where for each $s \in [S]$, left hand side of the constraint can be treated as a subproblem.

8.1. Cutting-Surface Algorithm for CS-DR-CCFP

Algorithm 2 can be leveraged to optimally solve (37) using a cutting surface algorithm such as the one developed in Luo & Mehrotra (2019). This algorithm solves a master problem with a fnite number of cuts, and uses a cut generation oracle. Let t be the master iteration number, $\Xi^t \subseteq \Xi$ be a discrete set containing t previously generated elements and $\xi^k \in \Xi^t$ for $k \in [t]$. The master problem is:

$$\begin{aligned} \min_{x, \nu, \mu} \quad & \tau\mu + \frac{1}{S} \sum_{s \in [S]} \nu_s \\ & ||\xi^k - \xi^s||_1 \mu + \nu_s \geq h(x, \xi^k), \; \forall k \in [t], s \in [S] \\ & x \in \mathcal{X}, \;\; \mu \geq 0 \end{aligned} \tag{38}$$

Master problems (38) can be solved using Algorithm 1. Let (x^k, μ^k, ν_s^k) be the solution of the master problem at the kth iteration. The cut generation subproblem for the finite resource allocation and Cobb-Douglas subproblems are as follows.

Equitable Resource Allocation Oracle

$$\max_{(\tilde{b},\tilde{d})\in\Xi} \quad \sum_{j=1}^{n} \left| \frac{\sum_{i=1}^{m} \tilde{b}_{ij} x_{ij}^{k}}{\sum_{i=1}^{m}\sum_{j=1}^{n} \tilde{b}_{ij} x_{ij}^{k}} - \frac{1}{n} \right| - \mu^k ||(\tilde{b},\tilde{d}) - (b^s, d^s)||_1 \tag{39}$$

Cobb-Douglas Oracle

$$\min_{\tilde{a},\tilde{c}\in\Xi} \quad \frac{\tilde{a}_0 \prod_{i=1}^{n} {x_i^k}^{\tilde{a}_i}}{\sum_{i=1}^{n} \tilde{c}_i x_i^k + \tilde{c}_0} + \mu^k ||(\tilde{a},\tilde{c}) - (a^s, c^s)||_1. \tag{40}$$

We note that S such oracle generation subproblems are solved at each master iteration. Using the solution of these problems we add at most S cuts $h(x,\xi^{t+1}) - \mu||\xi^{t+1} - \xi^s||_1 - \nu_s \leq 0, s \in [S]$. Such a cutting surface algorithm can be used to generate an ϵ-optimal solution. This result follows from Theorem 6.1 of Luo & Mehrotra (2019) assuming that the oracle problems are solved to $\epsilon/2$-optimality. Note that $h(x,\xi)$ is continuous and thus the assumptions of Theorem 6.1 of Luo & Mehrotra (2019) are satisfied. We, however, point out that the oracle generation subproblems in the resource allocation and Cobb-Douglas cases have a mixed fractional-convex structure, and the development of efficient algorithms for solving such problems requires additional research.

Acknowledgments

We would like to acknowledge NSF grant CMMI-1763035, which provided partial support for this work. The data that support the findings of this study are available from the corresponding author, SM, upon reasonable request. The problem generation is described within the article. The authors would like to thank anonymous referees and the journal's area editor for many helpful suggestions. Particularly, the proofs of sample average approximation results for the stochastic fractional programs were added and improved following referee suggestions.

References

Aggarwal, S., & Sharma, I. (1970). Maximization of the transmission rate of a discrete, constant channel. *Unternehmensforschung*, *14*, 152–155.

Benson, H. (2002a). Global optimization algorithm for the nonlinear sum of ratios problem. *Journal of Optimization Theory and Applications*, *112*, 1–29.

Benson, H. P. (2001). Global optimization of nonlinear sums of ratios. *Journal of Mathematical Analysis and Applications*, *263*, 301–315.

Benson, H. P. (2002b). Using concave envelopes to globally solve the nonlinear sum of ratios problem. *Journal of Global Optimization*, *22*, 343–364.

Bertsimas, D., Doan, X. V., Natarajan, K., & Teo, C.-P. (2010). Models for minimax stochastic linear optimization problems with risk aversion. *Mathematics of Operations Research*, *35*, 580–602.

Bradley, S. P., & Frey Jr, S. C. (1974). Fractional programming with homogeneous functions. *Operations Research*, *22*, 350–357.

Charnes, A., & Cooper, W. W. (1962). Programming with linear fractional functionals. *Naval Research Logistics Quarterly*, *9*, 181–186.

Cobb, C. W., & Douglas, P. H. (1928). A theory of production. *The American Economic Review*, *18*, 139–165.

Delage, E., & Ye, Y. (2010). Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, *58*, 595–612.

Dur, M., Horst, R., & Van Thoai, N. (2001). Solving sum-of-ratios fractional programs using efficient points. *Optimization*, *49*, 447–466.

Filipe, J., & Adams, F. G. (2005). The estimation of the Cobb-Douglas function: a retrospective view. *Eastern Economic Journal*, *31*, 427–445.

Freund, R. W., & Jarre, F. (2001). Solving the sum-of-ratios problem by an interior-point method. *Journal of Global Optimization*, *19*, 83–102.

Gruzdeva, T. V., & Strekalovsky, A. S. (2018). On solving the sum-of-ratios problem. *Applied Mathematics and Computation*, *318*, 260–269.

Hodgson, T. J., & Lowe, T. J. (1982). Production lot sizing with material-handling cost considerations. *IIE Transactions*, *14*, 44–51.

Hu, Y., Yu, C. K. W., & Yang, X. (2019). Incremental quasi-subgradient methods for minimizing the sum of quasi-convex functions. *Journal of Global Optimization*, *75*, 1003–1028.

Jiao, H., & Liu, S. (2017). An efficient algorithm for quadratic sum-of-ratios fractional programs problem. *Numerical Functional Analysis and Optimization*, *38*, 1426–1445.

Kim, C., & Mehrotra, S. (2021). Solution approaches to linear fractional programming and its stochastic generalizations using second order cone approximations. *SIAM Journal on Optimization*, *31*, 945–971.

Konno, H., Kuno, T., & Yajima, Y. (1994). Global minimization of a generalized convex multiplicative function. *Journal of Global Optimization*, *4*, 47–62.

Luo, F., & Mehrotra, S. (2019). Decomposition algorithm for distributionally robust optimization using Wasserstein metric with an application to a class of regression models. *European Journal of Operational Research*, *278*, 20–35.

Luo, F., & Mehrotra, S. (2020). Distributionally robust optimization with decision dependent ambiguity sets. *Optimization Letters*, *14*, 2565–2594.

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. *Mathematical Programming*, *10*, 147–175.

Mehrotra, S., & Papp, D. (2014). A cutting surface algorithm for semi-infinite convex programming with an application to moment robust optimization. *SIAM Journal on Optimization*, *24*, 1670–1697.

Meister, B., & Oettli, W. (1967). On the capacity of a discrete, constant channel. *Information and Control*, *11*, 341–351.

Rao, M. (1971). Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, *66*, 622–626.

Shapiro, A. (1991). Asymptotic analysis of stochastic rograms. *Annals of Operations Research*, *30*, 169–186.

Stancu-Minasian, I. M. (1997). *Fractional Programming: Theory, Methods and Applications* volume 409. Springer Netherlands.

Wang, Y., & Liang, Z. (2005). A deterministic global optimization algorithm for generalized geometric programming. *Applied Mathematics and Computation*, *168*, 722–737.

Wiesemann, W., Kuhn, D., & Sim, M. (2014). Distributionally robust convex optimization. *Operations Research*, *62*, 1358–1376.

Ziemba, W. T., Parkan, C., & Brooks-Hill, R. (2013). Calculation of investment portfolios with risk free borrowing and lending. In *Handbook of the Fundamentals of Financial Decision Making: Part I* (pp. 375–388). World Scientific.

Appendix A. Proofs

Proof. of the Proposition 3.1: Let us consider the first case. If $\sigma = 1$, we have $0 < \hat{d}_k$ by (A5). For $\sigma = -1$, from $g_k(x^*) < 0$, we have $0 < \hat{d}_k$. Therefore, regardless of the value of σ, $0 < \hat{d}_k$ holds. Based on the construction, since $\hat{\gamma}_k + \hat{d}_k = 2\hat{w}_k$, $\hat{\gamma}_k - \hat{d}_k = 2\hat{v}_k$, and $\hat{c}_k + \hat{v}_k^2 \leq \hat{w}_k^2$ which indicate $\hat{c}_k \leq \hat{d}_k\hat{\gamma}_k$, we have $\hat{c}_k/\hat{d}_k \leq \hat{\gamma}_k$ due to $\hat{d}_k > 0$.

Also, if $\sigma = 1$, we have $\hat{\gamma}_k = f_k(x^*)/g_k(x^*) \leq c_k^*/d_k^* = \gamma_k^*$ due to $f_k(x^*) > 0$ and $g_k(x^*) > 0$. On the other hand, for $\sigma = -1$, we obtain $\hat{\gamma}_k = f_k(x^*)/-g_k(x^*) \leq f_k(x^*)/d_k^* \leq c_k^*/d_k^* = \gamma_k^*$ since $f_k(x^*) < 0$ and $0 < -g_k(x^*) \leq d_k^*$. Since $\hat{\gamma}_k \leq \gamma_k^*$, we have $H^T\pi^* \geq \hat{\gamma}_k$ and $P\hat{\gamma} \leq \theta^*\mathbb{1}_J$ due to $P \geq 0$. Therefore, $(x^*, \hat{c}, \hat{d}, \theta^*, \hat{\gamma}, \pi^*, \hat{w}, \hat{v})$ is feasible to (17).

Suppose that $(x^*, \hat{c}, \hat{d}, \theta^*, \hat{\gamma}, \pi^*, \hat{w}, \hat{v})$ is not an optimal solution to (17). Let an optimal solution to (17) instead be $(\bar{x}, \bar{c}, \bar{d}, \bar{\theta}, \bar{\gamma}, \bar{\pi}, \bar{w}, \bar{v})$ such that $\bar{\theta} < \theta^*$. From that $\bar{\gamma}_k + \bar{d}_k = 2\bar{w}_k$, $\bar{\gamma}_k - \bar{d}_k = 2\bar{v}_k$, and $\bar{c}_k + \bar{v}_k^2 \leq \bar{w}_k^2$, we have $\bar{c}_k \leq \bar{d}_k\bar{\gamma}_k$. Since

$0 < \beta_k \leq \bar{d}_k$ if $\sigma = 1$ and $\bar{d}_k > 0$ due to $0 < -g_k(\bar{x}) \leq \bar{d}_k$ if $\sigma = -1$, we obtain $\bar{c}_k/\bar{d}_k \leq \bar{\gamma}_k$. Therefore, $(\bar{x}, \bar{c}, \bar{d}, \bar{\theta}, \bar{\gamma}, \bar{\pi})$ is feasible to (9) with the objective value of $\bar{\theta} < \theta^*$, which contradicts the fact that $(x^*, c^*, d^*, \theta^*, \gamma^*, \pi^*)$ is an optimal solution to (9). Thus, $(x^*, \hat{c}, \hat{d}, \theta^*, \hat{\gamma}, \pi^*, \hat{w}, \hat{v})$ is optimal to (17).

To prove the second statement, suppose that $(x^*, c^*, d^*, \theta^*, \gamma^*, \pi^*)$ is not an optimal solution to (9). Let $(\bar{x}, \bar{c}, \bar{d}, \bar{\theta}, \bar{\gamma}, \bar{\pi})$ be an optimal solution to (9) such that $\bar{\theta} < \theta^*$. Then, by the first part of the above argument, we can construct a feasible solution to (17), which has the objective value of $\bar{\theta}$. This contradicts the fact that $(x^*, c^*, d^*, \theta^*, \gamma^*, \pi^*)$ is an optimal solution to (9). Therefore, $(x^*, c^*, d^*, \theta^*, \gamma^*, \pi^*)$ should be optimal to (9) $\square$

Appendix B. Modified Adaptive Branch-and-Bound Algorithm

Algorithm 3 SOC-B-Efficient

1: optimality tolerance: $\epsilon > 0$
2: compute bounds on w_k, d_k and construct an initial hyper-rectangle B_0
3: solve (22) with $B = B_0$ and obtain $(\bar{x}(B_0), \bar{\vartheta}(B_0))$
4: compute $\psi(\bar{x}(B_0))$ by (23) and let $(x^0_{\text{CB}}, \theta^0_{\text{CB}}) \leftarrow (\bar{x}(B_0), \psi(\bar{x}(B_0)))$
5: let $\vartheta^t_{\text{CB}} \leftarrow \theta^0_{\text{CB}}$, $t \leftarrow 0$, $T_0 \leftarrow \{(B_0,\ \bar{\vartheta}(B_0))\}$, $tree2 \leftarrow 0$; $\zeta \leftarrow 0.3$
6: **while** true **do**
7:     **if** $T_t \neq \emptyset$ **then**
8:         find B_t such that $\bar{\vartheta}(B_t) = \min_{(B,\bar{\vartheta}(B)) \in T_t} \vartheta(B)$ and let $\bar{\vartheta}^t \leftarrow \bar{\vartheta}(B_t)$
9:         $T_{t+1} \leftarrow T_t \setminus \{(B_t, \bar{\vartheta}(B_t))\}$
10:     **else if** $\tilde{T} \neq \emptyset$ **then**
11:         **for** $(B, \bar{\vartheta}(B)) \in \tilde{T}$ **do**
12:             **if** $\bar{\vartheta}(B) \leq \theta^t_{\text{CB}}$ **then**
13:                 $T_{t+1} \leftarrow (B, \bar{\vartheta}(B))$
14:             **end if**
15:         **end for**
16:         $tree2 \leftarrow 1$; $\quad t \leftarrow t + 1$
17:         **Go** to next iteration if Line 19 is False
18:     **end if**
19:     **if** $\vartheta^t_{\text{CB}} - \bar{\vartheta}^t / |\vartheta^t_{\text{CB}}| < \epsilon$ **then**
20:         **return** x^t_{CB} and θ^t_{CB}
21:     **else**
22:         let $(x^{t+1}_{\text{CB}}, \theta^{t+1}_{\text{CB}}) \leftarrow (x^t_{\text{CB}}, \theta^t_{\text{CB}})$
23:         find $k_t = \arg\max_{k \in [K]} (\mathrm{w}^{b,t}_k - \mathrm{w}^{a,t}_k)^2 / d^m_k$ and let B'_t, B''_t as (24), (25)
24:         $flag = 0$
25:         **for** $B \in \{B'_t, B''_t\}$ **do**
26:             Solve LP relaxation of (22) with B to obtain $(\bar{x}^{\text{LP}}(B), \bar{\vartheta}^{\text{LP}}(B))$
27:             **if** $\theta^{t+1}_{\text{CB}} < \bar{\vartheta}^{\text{LP}}(B)$ **then**
28:                 fathom B; $\quad flag \leftarrow flag + 1$
29:             **else if** $\theta^{t+1}_{\text{CB}} \geq \bar{\vartheta}^{\text{LP}}(B)$ & $flag\ is\ 1$ & $\mathrm{w}^{b,t}_{k_t} - (\mathrm{w}^{a,t}_{k_t} + \mathrm{w}^{b,t}_{k_t})/2 \leq \zeta\{\mathrm{w}^M_{k_t} - (\mathrm{w}^m_{k_t} + \mathrm{w}^{b,t}_{k_t})/2\}$ & $tree2\ is\ 0$ **then**
30:                 $\tilde{T} \leftarrow \{(B''_t, \bar{\vartheta}(B''_t))\}$
31:             **else**
32:                 Solve (22) with B to obtain $(\bar{x}(B), \bar{\vartheta}(B))$ and (23) for $\psi(\bar{x}(B))$
33:                 **if** $\psi(\bar{x}(B)) < \theta^{t+1}_{\text{CB}}$ **then**
34:                     update $(x^{t+1}_{\text{CB}}, \theta^{t+1}_{\text{CB}}) \leftarrow (\bar{x}(B), \psi(\bar{x}(B)))$; $\vartheta^{t+1}_{\text{CB}} \leftarrow \theta^{t+1}_{\text{CB}}$
35:                 **end if**
36:                 $T_{t+1} \leftarrow T_{t+1} \cup \{(B, \bar{\vartheta}(B))\}$
37:             **end if**
38:         **end for**
39:     **end if**
40:     $t \leftarrow t + 1$
41: **end while**