Sum of Ratios Optimization Using a New Variant of Dinkelbach's Algorithm

Sahar Seidi Khorramabadi

A Thesis Submitted to the Faculty of Graduate Studies in

Partial Fulfillment of the Requirements for the Degree of Master of Applied Science

Graduate Program in Electrical and Computer Engineering

York University

Toronto, Ontario

May 2021

Abstract

The sum of ratios optimization problem appears in many different communications applications and therefore is an important topic to study. The classic power control problem in a communication link is defined as maximizing the data rate in a scenario with multiple users that can be modeled as a sum of the ratios optimization problem. The Dinkelbach's method has been extensively used in optimizing single ratios, but for many years, it was believed that no generalization to Dinkelbach's algorithm for the sum of ratios problem was possible, and a few attempts were proven wrong later. In this research, we propose a new generalization to the Dinkelbach's technique tailored to the sum of functions of ratios problem along with developing a new way to update the auxiliary variable in the equivalent parametric program. The new variant of Dinkelbach's algorithm is proved to be a fast and accurate technique to solve the problem of the sum of non-decreasing functions of ratios with concave numerators and convex denominators.

Contents

List of Figures

List of Tables

Chapter One: Fractional Programs, Background and Classic Approaches

Optimization is important to decision-making problems in areas like engineering and economics. Choosing between different options usually realizes the process of decision making, and the metric to measure the efficiency of our choices is called the objective function. A good example in communications engineering is optimizing the energy consumption in a wireless network that can drastically change the level of pollutions created as a result. The best decision is usually made with respect to some constraints on the parameters to choose, and as a result, there is usually a feasible set for the parameters that are used in an objective function [1].

There are many applications in engineering that could be modeled with liner programs such as optimizing a convex problem. Additionally, cases with non-linear objective functions are found in both engineering and economics problems. A special case of non-linear programming is when we have one or multiple ratios that need to be optimized. These problems are categorized as Fractional Programming problems. One early application of fractional programs is a model developed by Von Neumann for an economy in 1937 [2]. The authors proposed an equivalent objective function to the non-convex program and solved it. Aside from a few papers, the study of fractional programs did not start until 1960's with the work published by Charnes and Cooper [3]. Optimization problems also appear in various areas of engineering, including communications. These applications include both linear and non-linear objective functions. The general form of a fractional program, according to Charnes and Cooper [3], can be written as follows:

$$\min \quad f(x) = \sum_{i=1}^{p} \frac{h_i(x)}{g_i(x)} \tag{1.1}$$

$$\text{subject to } x \in \mathrm{X}$$

wherein the case of i =1 there is a single ratio to optimize and that forms a classic fractional program. Here, both functions in the numerator and denominator are real-valued and finite on a feasible set of X. Since these ratios are, in general, non-convex, we can not use simple optimization toolboxes or algorithms to optimize them, and instead, we need to find ways to transform the problem into an equivalent convex problem. This has been a prevalent topic in the optimization field for years and is still an open problem. Based on the nature of the numerator, denominator, and constraints, we can have a broader classification of fractional programming problems that are discussed below [4].

1.1 Classification of Fractional Programming Problems

Here, different categories of fractional programs are introduced according to [5]:

1.1.1 Single Ratio Linear Fractional Programming

This problem is defined as below:

$$\min \quad \frac{\mathrm{h(x)}}{\mathrm{g(x)}} \tag{1.2}$$

$$\mathrm{x} \in \mathrm{X}$$

$$\mathrm{g(x)} > 0 \text{ for all } \mathrm{x} \in \mathrm{X} \text{ and } \mathrm{l_k}: \mathrm{x} \rightarrow \mathrm{R}\ \mathrm{k} = 1: \mathrm{n\ a}$$

In this problem, both the numerator and denominator are affine (linear + constant) functions.

1.1.2 Single Ratio Fractional Programming

This problem can be defined as:

$$\min \frac{h(x)}{g(x)} \tag{1.3}$$

$$x \in X$$

where h and g are real-valued functions which are finite on X and $g(x) > 0 \; for \; all \; x \in X$ and X is a non-empty closed feasible region. This category also includes the problems from the first group

1.1.3 Single Ratio Quadratic Fractional Programming

This kind of problem is given by:

$$\min \frac{h(x)}{g(x)} \tag{1.4}$$

$$x \in X$$

Where h and g are quadratic functions and $g(x) > 0 \; for \; all \; x \in X \; \; and \; l_k : x \to R \; where \; k = 1 : n$ and $X = \{x : l_k(x) \leq 0, k = 1 : n\}$ are affine functions.

1.1. 4 Min-Max Fractional Programming

This problem is defined as:

$$\text{Min}_{x \in X} \text{Max}_{y \in Y} \frac{h(y, x)}{g(y, x)} \tag{1.5}$$

Where $X \in R^m$ and $Y \in R^n$ are non-empty closed sets.

1.1.5 Sum of Ratios Fractional Programming

This type of problem is given by:

$$\min \sum_{i=1}^{p} \frac{h_i(x)}{g_i(x)} \tag{1.6}$$

$$x \in X$$

Where $g_i(x) > 0\ for\ all\ i$ and $x \in X$.

1.1.6 Multi-Objective Fractional Programming

In this problem, we have multiple ratios that need to be optimized simultaneously as below:

$$\min \{\frac{h_1(x)}{g_1(x)}, \dots, \frac{h_p(x)}{g_p(x)}\} \tag{1.7}$$

$$x \in X$$

Where $g_i(x) > 0\ for\ all\ i$ and $x \in X$.

1.2 Classic Approaches to Solve Fractional Programming Problems

First, we consider the single ratio fractional programs, which are a broad category of fractional programs. Given a non-empty constraint set, we can define a single ratio optimization problem as below:

$$\min \quad \frac{h(x)}{g(x)} \tag{1.8}$$

$$\text{Subject to } x \in X$$

The problem above is non-convex in general. The most straightforward approach to deal with nonconvexity is to find a way to decouple the numerator and the denominator and transform the problem into a convex optimization case. There are two classic methods to do so. The first method was developed by Charnes-Cooper as an initial attempt to solve fractional programs, and the second one was proposed by Dinkelbach [5][6]. A detailed description of these two algorithms is presented below.

1.2.1 Charnes-Cooper Method

Based on the work presented in [3] and [5], this method uses a simple change of variables to transform the non-convex program to a convex equivalent objective function. Two new variables are introduced based on the original problem in (1.8):

$$z = \frac{1}{g(x)} \;, \qquad q = \frac{x}{g(x)} \tag{1.9}$$

Thus, the single ratio problem is reformulated as below:

$$\begin{aligned} &\min \quad z \times h\left(\frac{q}{z}\right) \\ &\text{subject to } z \times g\left(\frac{q}{z}\right) \leq 1 \\ &z \in Z \\ &q \in Q \end{aligned} \tag{1.10}$$

According to this change of variables, the objective function is no longer fractional and can be optimized by an appropriate optimization method. The introduced change of variables decouples the numerator and denominator and moves the denominator to the constraints. After solving the problem for the new variables, the value of the original variable x can be obtained using (1.9). As can be seen, this transform adds new constraints to the problem, which may not be favorable in all cases. This method was first proposed only for affine fractional programs by Charnes-Cooper and was later extended to the case of concave-convex by Schaible [5].

1.2.2 Dinkelbach's Method

This method was first introduced by Werner Dinkelbach in 1967 in [6] with a super-linear (faster than linear) convergence rate based on introducing a new auxiliary variable to decouple the numerator and the denominator. The objective function is modeled with an equivalent objective that is convex for fixed *y*:

$$\min \quad h(x) - y \times g(x) \tag{1.11}$$

$$\text{Subject to } x \in X$$

$$y_t = \frac{h(x_{t-1})}{g(x_{t-1})}$$

wherein the subscript *t* is the iteration index, and it is proved that in each iteration, with a fixed auxiliary variable, we only need to solve a convex problem. When the original single ratio problem is a concave-convex problem, the algorithm converges to the global optimum. The auxiliary variable *y* is updated in each iteration, and an equivalent convex objective is optimized for a fixed *y*. The main advantage of this algorithm over the Charnes-Cooper method is that it does not add new constraints to the problem.

Although classic approaches prove to be efficient in solving the single ratio problems, it is difficult to generalize them to the case of multiple ratio programs. Therefore, they can only be effective when dealing with single ratio problems.

1.3 Objective of the Study

The objective of this research is to propose a new generalization to the Dinkelbach's technique tailored to the sum of functions of ratios optimization problem along with developing a new way to update the auxiliary variable in the equivalent parametric program. Optimizing the sum of ratios is of great importance in many wireless communication applications and is considered to be NP-hard [7]. Studying the literature in this area proves that for solving single ratio optimization problems, classic approaches like Charnes-Cooper and especially Dinkelbach's are more reliable and efficient [38]. However, these methods can not be directly extended to the multiple ratio case, and we need to develop new approaches for this class of optimization problems. The main work published in this area is based on a branch-and-bound algorithm which is an efficient but time-consuming process [17] [18] [19]. Other approaches like [37] and [38] seem to be more straightforward and have promising performances. In [37], the authors developed a new variant of Dinkelbach's algorithm suitable for the sum of ratios problem. Their approach works well for different examples, but as they mention in the conclusion section, it lacks effectiveness due to the large number of iterations needed to perform in the search process to find the auxiliary variable of the equivalent parametric program. This became a motivation for us to try to develop a new formula for the auxiliary variable so that the update can be performed without having to search the whole search space.

The sum of ratios optimization problem appears in many different communications applications and, therefore, is an important topic to study. The classic power control problem in a communication link can be represented with the problem of maximizing the data rate in a scenario with multiple users that can simply be modeled as a multiple ratio optimization problem. Furthermore, in beamforming design for a Multiple Input Multiple Output (MIMO) cellular network, a weighted sum rate over the beamforming vectors needs to be optimized, which is another application of multiple ratio fractional programming. The most popular application of sum-rate optimization in communications is the energy efficiency maximization in wireless links that has been broadly investigated for the case of a single link in the literature [38] and can be formulated as a multiple ratio fractional programs in case of multiple links.

The main contribution of this work is the introduction of a new update formula for the auxiliary variable q motivated by the original Dinkelbach's method. Having the algorithm defined as such, it can be easily seen that the original Dinkelbach's algorithm is a particular case of our new general form where $i = 1$ and the update rule only includes one numerator and denominator. One possible advantage of this new formulation is that it can be used for both affine and the quadratic (polynomial) sum of ratios without having to modify the algorithm. We propose to apply this new variant of Dinkelbach's algorithm to various numerical examples and conduct a comparative analysis with the obtained results of other algorithms mentioned in [22-32]. Additionally, a new approximative method is introduced for the general problem of sum of non-decreasing functions of ratios problem. An example of this problem is the weighted sum rate optimization, in which we aim to optimize the sum of logs of ratios. At the end, the application of fractional programming for the sum of functions of ratios in communication systems will be explored thoroughly with the proposed method, and relevant simulation results will be provided.

Chapter Two: Literature Review

In the previous chapter, the mathematical formulations of different fractional programming problems were discussed along with the classic solutions. In this chapter, the two main categories of fractional programs (single and multiple ratio programs) are investigated in more detail, and the more recent approaches to solve each group of problems is presented.

2.1 Single Ratio Case

The study of fractional programming problems with only a single ratio has primarily dominated the literature in the fractional programming field by 1980, with Dinkelbach's [3] and Charnes-Cooper's [6] approaches being the most promising methods to use. In [8], the author proposed a new combined algorithm to solve the single ratio fractional program such that both the objective function and the feasible region turn into DC (difference of two convex) functions. A new transform according to the equation (2.2) was proposed for the original problem defined as follows:

$$\max \ \frac{f(x)}{g(x)} \tag{2.1}$$

$$\text{s.t.} \quad x \in S$$

With λ introduced as an auxiliary parameter to define the following equivalent problem:

$$F(x,\lambda) := \begin{cases} \lambda f(\lambda^{-1}x) & \text{if } \lambda > 0, \\ 0 & \text{if } \lambda = 0, x = 0, \\ -\infty & \text{otherwise.} \end{cases} \tag{2.2}$$

The author proved the convexity of F in the case that the numerator is a convex function and, by means of a variable transformation as below, converted the problem to a DC optimization case:

$$\beta(x) = \frac{1}{g(x)}\ , \qquad y = x \times \beta(x) \tag{2.3}$$

$$F(y,\beta) = \beta \times f\left(\frac{y}{\beta}\right) = \beta \times f(x) = \frac{f(x)}{g(x)} \tag{2.4}$$

$$G(y,\beta) = \beta \times g\left(\frac{y}{\beta}\right) = \beta \times g(x) = 1$$

The equivalent DC problem can be written as follows:

$$\max\ F(y,\beta) \tag{2.5}$$

$$\text{s.t.}\quad G(y,\beta) - 1 \le 0$$

$$H(y,\beta) \le 0$$

$$\underline{\beta} \le \beta \le \overline{\beta}$$

With the following definitions:

Denote $\underline{\beta} =: 1/\max\{g(x) \mid x \in S\}$, $\overline{\beta} =: 1/\min\{g(x) \mid x \in S\}$. Then we see that:

$$S^H \subseteq \{(y,\beta) \in R^{n+1} \mid H(y,\beta) \le 0, \underline{\beta} \le \beta \le \overline{\beta}\} \cap \{(y,\beta) \mid G(y,\beta) - 1 \le 0\} \tag{2.6}$$

Using the above transformation, the author proved the convergence of the proposed algorithm and concluded that their method can achieve the desired results in a reasonably shorter time as compared with the original parametric method proposed by Dinkelbach.

Authors in [9] presented a new method to optimize a ratio in which both the numerator and denominator are convex, and at least one of them is in quadratic form. The first problem (P1) is given by:

$$\max \frac{x^T Q x}{x^T P x} \tag{2.7}$$

$$\text{Subject to } x \in X$$

wherein Q and P are both n$\times n$ positive semidefinite matrices of real numbers.

The second problem (P2) is defined as:

$$\max \frac{x^T Q x}{g(x)} \tag{2.8}$$

$$\text{Subject to } x \in X$$

The branch-and-bound algorithm solves the problem (P2) by solving the equivalent objective function as below:

$$\max f(y,t) \tag{2.9}$$

$$\text{s.t.} \quad (y,t) \in Y.T. \cap H$$

$$LY_j = \min y_j \tag{2.10}$$

$$\text{s.t.} \quad h_i(y) \leq 0, i = 1,2, \ldots q$$

$$L \leq W_y \leq U$$

and

$$UY_j = \max \; y_j \tag{2.11}$$

$$\text{s.t.} \quad h_i(y) \leq 0, i = 1,2, \dots q$$

$$L \leq W_y \leq U$$

For $L, U \epsilon R^n$.

For each j, the values of LY_j and UY_j are finite and each of them is the optimal value for a convex problem. Let us define the hyperrectangle H by $H = H_1 \times H_2 \times \dots . \times H_n$, where:

$$H_j = \{(y_i, t_i) \in R^2 | \, LY_j \leq y_j \leq UY_j, m \leq t_j \leq M\} \tag{2.12}$$

$$f_j(y_j, t_j) = \frac{\alpha_j y_j^2}{t_j}$$

And $f(y, t)$ is defined as below:

$$f(y, t) = \sum_{j=1}^{n} f_j(y_j, t_j) \tag{2.13}$$

and YT is given by any set of (y, t) that satisfies the following conditions:

$$h_i(y) \leq 0, i = 1,2, \dots . q \tag{2.14}$$

$$L \leq W_y \leq U$$

$$t_1 - h(y) \geq 0$$

$$t_1 = t_2 = \dots . = t_n$$

Using the above transform, the problem (P2) is globally solved by solving a sequence of convex programs provided that the number of variables is small. Similar algorithms were also proposed for the case of quadratic fractional programs using a generalized Dinkelbach method in [10]. Furthermore, authors in [11] and [12] presented a new algorithm to solve a group of linear fractional programming problems by transforming the problem to a linear program and then solving it by the simplex method.

2.2 Multiple Ratio Case

Reviewing the research published in the field of single ratio fractional programs proves that Dinkelbach's and Charnes-Cooper algorithms seem to have the best performance. However, a significant disadvantage of Dinkelbach's method is that it can not be extended to the case of multiple ratio programs (sum of ratios optimization) except in the case of the Min-Max problem [13]. In fact, in [14], authors tried to develop an extension to Dinkelbach's algorithm for the sum of ratios problem, but Falk and Palocsay [15] showed that their algorithm did not work in general.

A simple extension to Dinkelbach's was proposed in [16] based on presenting a new definition for the auxiliary parameter for the case with several ratios in which the update equation for the parameter is modified as below:

$$\lambda_{k+1} = \max \{\frac{f_i(x^k)}{g_i(x^k)}\} \tag{2.15}$$

And the new objective function is given by:

$$F(\lambda_k) = \inf\ [\max \{f_i\ (x) - \lambda_k g_i(x)\}] \tag{2.16}$$

The rest of the algorithm is similar to the original Dinkelbach's. The authors also presented a new algorithm and compared the convergence performance of both methods. This method seems to be converging very fast, but convergence to the global optimum point is not guaranteed, and in fact, in some cases, it converges to the local optimal solutions.

Other branch-and-bound algorithms were presented by Benson in [17], [18] and [19]. In [17] the following function was employed to solve the minimization of sum of ratios problem:

$$\psi(y,z) = \frac{\left[\sum_{i=1}^{p} y_i\left(\prod_{j=1,j\neq i}^{p} z_i\right)\right]^{1/p}}{\left(\prod_{i=1}^{p} z_i\right)^{1/p}} \tag{2.17}$$

where $y = (y_1, y_2, \ldots, y_p)$ and $z = (z_1, z_2, \ldots z_p)$ and W and H are defined as below:

$$W = \{(y,z) \in R^{2p} \mid y_i = f_i(x), z_i = g_i(x) \text{ for some } x \in S\} \tag{2.18}$$

$$H = \{(y,z) \in R^{2p} \mid 0 \leq y_i \leq \max_{x \in S} f_i(x), 0 \leq z_i \leq \max_{x \in S} g_i(x)\}$$

Thus, the following problem needs to be solved:

$$\min \{\psi(y,z) \mid (y,z) \in W\} \tag{2.19}$$

By branching on H and bounding performed on $H \cap W$, a lower bound of the numerator and an upper bound of the denominator are found. The authors proved the superiority of the proposed method in terms of finding the global optimal points, especially for the non-linear sum of ratios problem.

Two other approaches were proposed in [18] and [19] by Benson in order to solve problem (2.20) with a maximization objective. In [18] a new equivalent objective function is presented as follows:

$$\max\{\sum_{i=1}^{p}\frac{f_i(x)}{g_i(x)} \mid x \in S\} \tag{2.20}$$

$$\upsilon_H = \max \sum_{i=1}^{p}\frac{y_i}{z_i} \tag{2.21}$$

$$\begin{aligned} \text{s.t.} \quad & f_i(x) - y_i \geq 0, i = 1,2, \dots . p \\ & -g_i(x) + z_i \geq 0, i = 1,2, \dots . p \\ & x \in S, (y, z) \in H \end{aligned}$$

where H is a 2p dimensional box. The concave envelope of $\frac{y_i}{z_i}$ can be obtained by minimizing the following objective:

$$\min\{\frac{1}{\underline{z}_i}y_i - (\frac{\underline{y}_i}{\underline{z}_i\overline{z}_i})z_i + \frac{\underline{y}_i}{\underline{z}_i}, \frac{1}{\overline{z}_i}y_i - (\frac{\overline{y}_i}{\underline{z}_i\overline{z}_i})z_i + \frac{\overline{y}_i}{\underline{z}_i}\} \tag{2.22}$$

Using another branch-and-bound algorithm, authors proposed a global optimization method to solve the sum of ratios problem (2.20). The authors also proved that their method is computationally efficient and verified the results with numerical examples.

In [19], Benson suggested another equivalent objective to solve (2.20) as given follows:

$$\max \sum_{i=1}^{p} y_i f_i(x) \tag{2.23}$$

$$\text{s.t. } y_i g_i(x) - 1 \le 0, i = 1,2, \dots . p$$

$$x \in S, y \in \{y \in R^p \mid \frac{1}{\max_{x \in S} g_i(x)} \le y_i \le \frac{1}{\min_{x \in S} g_i(x)}\}$$

The author backed up this new algorithm with numerical examples and suggested that solving the presented equivalent objective is more computationally efficient and can be used to globally solve non-linear sum of ratios problems. However, an accurate computational analysis is still lacking in both [18] and [19].

Authors in [20] considered a general case of problem (2.20) where both the numerator and the denominator are DC functions. The original problem was converted to a linear programming problem and solved using an algorithm proposed in [21].

Another branch-and-bound algorithm to solve the sum of ratios problem was suggested by Dur et al. in [22]. Provided that the following assumptions are fulfilled, their algorithm can solve the problem (2.20):

(i) The functions f_i are positive concave, and the functions g_i are convex.

(ii) All functions are affine.

Assume that for $i = 1, \dots p$ the following bounds can be computed:

$$l_i^0 \le \min\{\frac{f_i(x)}{g_i(x)} \mid x \in P\} \tag{2.24}$$

$$u_i^0 \geq \max\{\frac{f_i(x)}{g_i(x)} | \ x \ \in \ P\}$$

After introducing the new variable y as $y = (y_1, \dots . y_p)$, and the rectangle $Y^0 = \{y \in \ R^p : l^0 \leq y \leq \ u^0\}$, it can be seen that the problem (2.20) is equivalent to the following problem:

$$\max \sum_{i=1}^{p} y_i \tag{2.25}$$

$$\text{s. t.} \ \ f_i(x) - y_i \times g_i(x) \geq 0, i = 1, \dots . p$$

$$x \ \in P, y \ \in \ Y^0$$

The authors then presented a branch-and-bound algorithm to solve the equivalent problem above. For the case of maximizing the sum of affine ratios, the Lagrange-dual of the problem is reduced to a linear program and can be easily solved. To compute lower bounds, they suggested using the efficient points of associated multiple objective problems.

By considering the two following problems, the authors in [23] proposed an equivalent problem to solve the nonlinear sum of ratios problem:

$$(\mathbf{P}): \min \quad h(x) = \sum_{j=1}^{p} h_j(x) = \sum_{j=1}^{p} \frac{n_j(x)}{d_j(x)} \tag{2.26}$$

$$\text{s.t.} \ \ g_m(x) \leq 0, m = 1, \dots . M$$

$$x \ \in \ X \ = \ \{0 < x_i^l \leq x_i \leq x_i^u, i = 1, \dots, N\}$$

They assumed that all the numerators and denominators were polynomials with real constant coefficients. An equivalent to problem P is given by:

$$(\mathbf{P}'):\ \min \quad h(x) = \sum_{j=1}^{p} h_j(x) = \sum_{j=1}^{p} \frac{t_j}{s_j} \tag{2.27}$$

$$\text{s.t.} \quad t_j - n_j(x) \leq 0, j = 1, \dots . p$$

$$d_j(x) - s_j \leq 0, j = 1, \dots \dots , p$$

$$g_m(x) \leq 0, m = 1, \dots M$$

$$x \in X = 0 < x_i^l \leq x_i \leq x_i^u, i = 1, \dots , N$$

$$(t, s) \in \Omega = \{(t, s) \in R^{2p} \mid 0 < l_j \leq t_j \leq u_j, 0 < L_j \leq s_j \leq U_j\}$$

Where $n_j(x) = t_j$ and $\Omega = \{(t, s) \in R^{2p} | 0 \leq l_j \leq t_j \leq u_j, 0 \leq L_j \leq s_j \leq U_j\}$. Then, using an exponential change of variables, they rewrote the problem as below:

$$(\mathbf{Q}): \quad \min \ f_0(y) \tag{2.28}$$

$$\text{s.t.} \ f_m(y) \leq 0, m = 1, \dots , 2p + M$$

$$y \in E = \underline{y_j} \leq y_j \leq \overline{y_j}, j = 1, \dots . , 2p + N$$

$$= \{ \ln(x_i^l) \leq z_i \leq \ln(x_i^u), i = 1, \dots . N,$$

$$\ln(l_j) \leq \eta_j \leq \ln(u_j), j = 1, \dots p$$

$$\ln(L_j) \leq \xi_j \leq \ln(U_j), j = 1, \dots p\}$$

where $x_i = \exp(z_i)$, $t_j = \exp(\eta_j)$, $s_j = \exp(\xi_j)$.

Introducing the concept of Linear Bounding Functions, the authors found the linear relaxation of problem Q and develop a branch-and-bound algorithm to solve problem Q. The lower bounding problem is transformed into a linear programming problem and solved by the simplex method. It was claimed that the proposed method was efficient when optimizing the non-linear sum of ratios problem over a non-convex feasible region.

A new algorithm to solve the sum of linear 1-D and 2-D fractional functions was developed in [24]. The original problem was given by:

$$\max \quad f(x) = \sum_{i=1}^{m} \frac{n_i(x)}{d_i(x)} \tag{2.29}$$

$$x \in S$$

If the problem is d-dimensional, and $\bar{x} = (x_1, \ldots . x_d)^T$, the sum of linear fractions is transformed to an m-dimensional equivalent problem using an iterative algorithm. Each ratio $r_i(\bar{x}) = \frac{n_i(\bar{x})}{d_i(\bar{x})}$ is mapped to a corresponding dimension r_i in the new space. Therefore, the equivalent objective function is defined as follows:

$$F(\bar{r}) = \sum_{i=1}^{m} r_i \tag{2.30}$$

Using an iterative algorithm, both the lower and upper bounds of each dimension r_i were computed, trying to minimize the gap between the two values in each iteration. An optimal solution was obtained when the upper bound matched the lower bound. Then they proposed a linear parametric programming approach to solve the offline 2-D ratio queries problem, developing the method for the single ratio case and then generalizing it to the case with M ratios. Considering that there are N linear constraints in the original problem, the authors suggested a way to process all the calculations in $O((N + M)LogN)$ time, which is an advantage compared to the conventional methods. They also claimed that their method was more straightforward to implement. The parametric approach is similar to that of Dinkelbach's, and it is generalized to the case of multiple ratios. The authors also suggested that some computational geometry problems can be transformed to the sum of linear ratios problem and solved using their algorithm. These problems include the problem of computing length-optimal supports for a simple non-convex polygon that appears in layered manufacturing that can be reduced to a 1-D sum of linear ratios problem. The results were verified by solving numerical examples.

Another branch-and-bound algorithm was presented in [25] to globally solve the sum of ratios problem with coefficients. The original problem was given by:

$$(\mathbf{P}): \quad \max \ f(x) = \sum_{i=1}^{m} c_i \frac{n_i(x)}{d_i(x)} \tag{2.31}$$

$$\text{s.t.} \quad x \in X$$

Where $m \geq 2$ and $n_i(x), d_i(x)$ are finite affine functions on R^n such that $n_i(x) \geq 0$ and $d_i(x) > 0$ for all $x \in X = \{x|Ax \leq b\}$ and c_j are real constant coefficients. The authors noted that this kind of problem had many applications in finance etc., and a significant characteristic of it was having multiple local optimums that make it difficult to deal with. The authors proposed an equivalent non-convex problem to (2.31) and proved that it converged to the same solution. Assume that $c_i > 0$ for $i = 1, \ldots, T$ and $c_i < 0$ for $i = T+1, \ldots .m.$ $\overline{l_\iota} = \min n_i(x)$ and $\bar{u}_\iota = \max n_i(x)$, $\overline{L_\iota} = \min d_i(x)$ and $\overline{U_\iota} = \max d_i(x)$ for i=1…., m. Since $n_i(x), d_i(x)$ are affine functions, $\overline{l_\iota}, \bar{u}_\iota$, $\overline{L_\iota}, and$ $\overline{U_\iota}$ can be computed in a straightforward manner using linear programming.

Let $H = \{(t,s) \in R^{2m} | \overline{l_\iota} \leq t_i \leq \bar{u}_\iota, \overline{L_\iota} \leq s_i \leq \overline{U_\iota}, i = 1, \ldots, m\}$. The following equivalent program was proposed to be solved instead of problem P:

$$(\mathbf{Q}): \quad \upsilon_H = \max \ g(t,x) = \sum_{i=1}^{T} c_i \frac{t_i}{s_i} + \sum_{i=T+1}^{m} c_i \frac{t_i}{s_i} \tag{2.32}$$

$$\text{s.t.} \qquad n_i(x) - t_i \geq 0, i = 1, \ldots . T$$

$$n_i(x) - t_i \leq 0, i = T+1, \ldots . m$$

$$-d_i(x) + s_i \geq 0, i = 1, \ldots ., T$$

$$-d_i(x) + s_i \leq 0, i = T+1, \ldots ., m$$

$$x \in X, \quad (t,s) \in H$$

Then using the following theorem, the authors proved the convergence of the solutions of problems P and Q.

Theorem 1. If (x^*, t^*, s^*) is a global solution of problem (Q), then $t_i^*=n_i(x^*), s_i^* = d_i(x^*)$ and x^* is a global solution of problem (P). The converse is also correct. (Proof in [24]).

Therefore, using the linearization method, they obtained a linear relaxation equivalent of problem Q. In the algorithm, the branch-and-bound tree creates rectangular regions that belong to R^{2m}, where *m* is the number of ratios in the problem P. However, the branching process only takes place in R^m, rather than R^{2m}. By providing numerical examples, the authors verified the validity of their approach for problems with up to four ratios.

A deterministic global optimization approach to solve the linear sum of ratios problem was proposed in [26]. An equivalent objective function was derived by exploiting the characteristics of the constraints of the original problem. The problem is considered to have the following form:

$$\min \ f(x) = \sum_{j=1}^{p} h_j(x) = \sum_{j=1}^{p} \frac{c_{j0} + c_j^T x}{d_{j0} + d_j^T x} \tag{2.33}$$

$$\text{s.t.}\quad Ax \le b,\ \ x \in R^n$$

Where $A \epsilon R^{m\times n}$, $b \in R^m$ and c_{j0}, d_{j0} are all arbitrary real numbers and $c_j, d_j \in R^n$ and $c_{j0} + c_j^T x \ge 0$ and $d_{j0} + d_j^T x \ne 0$ for any $x \in \{x|Ax \le b\}, j = 1, \dots p$.

The authors then proposed a linear relaxation program as follows:

First, the following optimization problem is solved:

$$\underline{x_i} = \min\ \ x_i \tag{2.34}$$

$$\text{s.t.}\quad Ax \le b$$

$$\overline{x_i} = \max\ \ x_i$$

$$\text{s.t.}\quad Ax \le b$$

Then we get the initial interval vector $X^0 = \{x: \underline{x_i} \le x_i \le \overline{x_i}, i = 1, \dots n\}$. The equivalent problem is given by:

$$\min\ \ f(x) = \sum_{j=1}^{p} h_j(x) = \sum_{j=1}^{p} \frac{c_{j0} + c_j^T x}{d_{j0} + d_j^T x} \tag{2.35}$$

$$\text{s.t.}\quad Ax \le b,\ \ x \in X^0$$

By solving a sequence of linear relaxation programs, the upper and lower bound of the original problem were obtained, resulting in a globally optimum solution. The authors also verified their method by solving numerical examples.

Crouziex et al. presented a new algorithm to solve the generalized fractional programming problems in [27]. Based on their algorithm, the auxiliary variable is determined according to the value of the iterate in previous iterations and not the more recent one. The generalized fractional program is defined as below:

$$\lambda^{*} = \min_{x}[\max_{1\leq i\leq p}\frac{f_i(x)}{g_i(x)} : x \in C] \tag{2.36}$$

$$F_w(\lambda) = \min_{x}[\max_{1\leq i\leq p}\frac{f_i(x) - \lambda g_i(x)}{w_i} : x \in C] \tag{2.37}$$

$$= \min_{x,t}[t: f_i(x) - \lambda g_i(x) - tw_i \leq 0, 1 \leq i \leq p: x \in C, t \in R]$$

where $\emptyset \neq C \cap R^n$, $f_i: C \rightarrow R$ *and* $g_i: C \rightarrow (0, +\infty)$ for i = 1…p.

The authors proposed a new variant of the Dinkelbach's algorithms as follows:

- Start with $x_0 \in C$
- Take $\mu_0 = \max_{1\leq i\leq p}\frac{f_i(x_0)}{g_i(x_0)}$. Let k =1.
- Step k: Solve the problem: $\min_{x,t}$ $[t: f_i(x) - \mu_{k-1} g_i(x) - tw_i{}^{k-1} \leq 0, 1 \leq i \leq p: x \in C, t \in R]$
- Let (x_k, t_k) be an optimal solution to the problem above and: $\overline{\lambda_k} := \max_{1\leq i\leq p}\frac{f_i(x_k)}{g_i(x_k)}$

Stopping rule: If $t_k = 0$, then stop. $\lambda^{*} = \mu_{k-1}$ and x_k is an optimal solution of the generalized fractional program.

- If $t_k > 0$, then $lb < \mu_{k-1} < \lambda^{*}$

  Let $lb = \mu_{k-1}$

  If $\overline{\lambda_k} < \overline{\lambda_{k-1}}$, then $x_0 = x_k$ and go to step 0.

If $\overline{\lambda_k} \geq \overline{\lambda_{k-1}}$,then $x_0 = x_{k-1}$ and go to step 0.

- If $t_k < 0$, then $\lambda^* < \mu_{k-1} < \overline{\lambda_{k-1}}$

  Let $\mu_k = \max_{1 \leq i \leq p} \min_{0 \leq j \leq k} \frac{f_i(x_j)}{g_i(x_j)}$

  If $\mu_k \leq lb$, then $x_0 = x_k$ and go to step 0

  Otherwise, choose $w^k > 0$. Let k = k+1 and go to step k.

Using the above modifications to the Dinkelbach's algorithm, the authors proved that their new variant of the Dinkelbach's method is effective, especially when dealing with linear multiple ratios optimization problems.

Another branch-and-bound algorithm was presented by Chun-Feng and Pei-Ping [28] to solve the general linear fractional programming problem. The original problem was considered to have the following form:

$$(\mathbf{GFP}): \quad \max \quad g(x) = \sum_{i=1}^{p} \frac{\sum_{j=1}^{n} c_{ij}x_j + d_i}{\sum_{j=1}^{n} e_{ij}x_j + f_i} \qquad (2.38)$$

$$\text{s.t.} \quad Ax \leq b, \quad x \geq 0$$

The authors first presented a theorem that is the basis for their proposed global optimization algorithm as follows:

Theorem 2: Assume $\sum_{j=1}^{n} e_{ij}x_j + f_i \neq 0 \; for \; \forall x \in \Lambda, then \; \sum_{j=1}^{n} e_{ij}x_j + f_i > 0 \; or \; \sum_{j=1}^{n} e_{ij}x_j + f_i < 0$. See [28] for the proof.

Next, they transform the problem GFP to an equivalent problem EP given by:

$$EP = \max\ \varphi_0(x,y) = \sum_{i=1}^{p} y_i\left(\sum_{j=1}^{n} c_{ij}x_j + d_i\right) \tag{2.39}$$

$$s.t.\ \varphi_i(x,y) = y_i\left(\sum_{j=1}^{n} e_{ij}x_j + f_i\right) \le 1, i = 1, \dots, p$$

$$x \in \Lambda, y \in H^0$$

Where $\overline{l_i} = \min_{x\in\Lambda} \sum_{j=1}^{n} e_{ij}\, x_j + f_i, i = 1, \dots p$ and $H^0 = \{y \in R^p | l_i^0 \le y \le u_i^0, i = 1, \dots, p\}$ and $l_i^0 = \frac{1}{u_i}$ *and* $u_i^0 = \frac{1}{l_i}$.

The branching process is performed in R^p instead of R^n and iteratively divides the p-dimensional rectangle H^0 of problem $E.P.(H^0)$ into smaller sub rectangles of the same dimension. For each rectangle H, the upper bound process is applied to find the upper bound *U.B. (H)* for the optimal value v(H) of problem *E.P. (H)*. The upper bound can be found by solving a linear program in each iteration. The same procedure is taken to find the lower bounds of the problem, and the global solution is achieved when the upper and lower bounds converge. To verify the validity of their method, the authors provided some numerical examples of up to four ratios and showed that their algorithm can efficiently solve the sum of linear ratios problem.

Global optimization of non-linear sum of ratios problem is considered in [29] based on a new branch-and -bound algorithm. The general form of the problem is given by:

$$(\mathbf{P}):\ \min h(x) = \sum_{j=1}^{p} h_j(x) = \sum_{j=1}^{p} c_j \frac{n_j(x)}{d_j(x)} \tag{2.40}$$

$$s.t.\ g_m(x) \le 0, m = 1,2, \dots, M$$

$$x \in X^0 = \{x \in R^N | \underline{x} \leq x \leq \overline{x}\}$$

Where $n_j(x), d_j(x)$ and $g_m(x)$ are generalized multivariable polynomials. To globally solve the problem P, an equivalent non-convex problem is proposed as follows:

$$\min \ h(t,s) = \sum_{j=1}^{p} h_j(t,s) = \sum_{j=1}^{p} c_j \frac{t_j}{s_j} \tag{2.41}$$

$$t_j - n_j(x) \leq 0, j = 1, \dots, p$$

$$d_j(x) - s_j \leq 0, j = 1, \dots, p$$

$$g_m(x) \leq 0, m = 1, \dots, M$$

$$x \in X, (t,s) \in H^0$$

wherein it is assumed that $n_j(x) = t_j$, $d_j(x) = s_j$ and $H^0 = \{(t,s) \in R^{2p} | l_i \leq t_i \leq u_i, L_i \leq s_i \leq U_i, i = 1, \dots p\}$. Thus, using linear relaxation programming, the new problem is solved through a branching and bounding process. The most important factor in guaranteeing convergence to the global optimum solution is the method used for partitioning. The selection of the branching variable y_p is made according to the following rule:

Let $p = \arg\max\{\overline{y_i} - \underline{y_i}: 1, \dots 2p + N\}$ and partitioning y_p by bi-sectioning the interval $[\underline{y_p}, \overline{y_p}]$ into the subintervals $[\underline{y_p}, \frac{\underline{y_p} + \overline{y_p}}{2}]$ and $[\frac{\underline{y_p} + \overline{y_p}}{2}, \overline{y_p}]$. The authors verified the performance of the proposed method by a numerical example.

A new approach to globally solve the sum of ratios problem is proposed by Jong in [30] and [31]. The sum of ratios problem is formulated as follows:

$$\min \sum_{i=1}^{N} F_i(x) \tag{2.42}$$

$$\text{s.t.} \quad g_i(x) \leq 0, i = 1, \dots, m$$

$$x \in R^n$$

where $F_i(x) = \frac{f_i(x)}{h_i(x)}$ and all the numerators and denominators are continuously differentiable functions. It is also assumed that $f_i(x) \geq 0$ *and* $h_i(x) > 0$ in the feasible set. The proposed equivalent problem is given by:

$$\min \sum_{i=1}^{N} \beta_i \tag{2.43}$$

$$\text{s.t.} \quad F_i(x) \leq \beta_i, i = 1, \dots, N$$

$$g_i(x) \leq 0, i = 1, \dots, m$$

$$x \in R^n$$

The author then proved the optimality of the above problem by considering the Karush-Kuhn-Tucker (K.K.T.) conditions and the fact that the new equivalent problem is convex and can be solved by means of any convex programming techniques. They also proved that this new method benefited from a super-linear rate of convergence and backed this proof up by providing numerical examples and comparing the results with previously published results. Their method is applicable to both affine and the quadratic sum of ratios problems.

The following sum of linear ratios problem is considered in [32] and a branch and bound algorithm is proposed to convert the problem to a linear program:

$$\min \quad f(x) = \sum_{i=1}^{p} f_i(x) = \sum_{i=1}^{p} \frac{n_i(x)}{d_i(x)} \tag{2.44}$$

$$\text{s.t.} \quad Ax \leq b, \quad x \geq 0$$

Where A and b are real numbers. First, the authors constructed a rectangle $X = [l, u]$ which contained the feasible region of the solution, $l = [l_1, l_2, \dots, l_n]^T, u = [u_1, u_2, \dots, u_n]^T, l_j$ and u_j are the optimal solutions of the following linear programs respectively:

$$\min \quad l(x_j) = x_j \tag{2.45}$$

$$\text{s.t.} \quad Ax \leq b, \quad x \geq 0$$

$$\max \quad u(x_j) = x_j$$

$$\text{s.t.} \quad Ax \leq b, \quad x \geq 0$$

Then they solved the following linear programming problems:

$$\min \quad d_i(x) \tag{2.46}$$

$$\text{s.t.} \quad x \in D$$

$$\max \quad d_i(x)$$

$$\text{s.t.} \quad x \in D$$

The optimal solutions of (2.46) are x_i^1, x_i^2 and the optimal values are $\overline{l_\iota}, \overline{u_\iota}$. The set of current feasible solutions of (2.44) can be represented by:

$$H^0 = \{y \in R^p \mid l_i^0 \leq y_i \leq u_i^0, i = 1,2, \dots . p\} \tag{2.47}$$

$$y = (y_1, y_2, \dots, y_p)^T$$

Where $l_i^0 = \frac{1}{\overline{l_\iota}}$, $u_i^0 = \frac{1}{\overline{u_\iota}}$. Finally, the equivalent non-convex problem can be written as:

$$\mathrm{EP} = \min \quad \varphi_0(x, y) = \sum_{i=1}^{p} y_i\, n_i(x) = \sum_{i=1}^{p} y_i\Big(\sum_{j=1}^{n} c_{ij}x_j + d_i\Big) \tag{2.48}$$

$$\text{s.t.} \quad \varphi_i(x, y) = y_i d_i(x) = y_i\left(\sum_{j=1}^{n} e_{ij}x_j + r_i\right) \geq 1, i = 1, \dots., p$$

$$x \in D \cap X, y \in H^0$$

The authors then proved the equivalency of the problem above with the original problem and gave a linear relaxation method to solve (2.48). They also proposed a new branching rule to choose the sub rectangles in the branching process. They proved the superiority of their method by conducting numerical experiments and comparing the results with those of [28], concluded that their algorithm can achieve feasible and efficient optimal solutions.

Authors in [33] considered maximization of sum of linear fractions as below:

$$\max f(x) := \sum_{i=1}^{p} \frac{n_i^T x + a_i}{d_i^T x + b_i} \tag{2.49}$$

$$\text{s.t.} \quad x \in X := \{x: Ax \leq c\}$$

wherein A and c are real numbers. By defining a linear relaxation for the problem above, the following equivalent problem is proposed that can be solved instead of problem (2.49):

$$\max \sum_{i=1}^{p} (n_i^T y^i + a_i z_i) \tag{2.50}$$

$$\text{s.t.} \quad d_i^T y^i + b_i z_i = 1, i = 1, \dots, p$$

$$Ay^i - cz_i \leq 0, i = 1, \ldots, p$$

$$\frac{1}{\beta_i} \leq z_i \leq \frac{1}{\alpha_i}, i = 1, \ldots p$$

$$y^i z_j = y^j z_i, i, j = 1, \ldots, p$$

where $\alpha_i = \min[d_i^T x + b_i | x \in X]$ and $\beta_i = \max[d_i^T x + b_i | x \in X]$.

The authors claimed that their algorithm was more efficient when the number of ratios is large, and the problem has lower dimensions. Compared to the conventional branch-and-bound algorithm, this method benefits from faster convergence and less CPU time, especially in the case of a more significant number of ratios ($p \geq 15$). However, the authors only provided the benchmark results of numerical examples of four ratios.

A global optimization algorithm is proposed in [34] for sum of generalized polynomial ratios based on a branch-and-bound algorithm using an exponential change of variables. The authors considered the following problem:

$$\min \quad h_0(x) = \sum_{j=1}^{p} \frac{n_j(x)}{d_j(x)} \tag{2.51}$$

$$\text{s.t.} \quad h_m(x) \leq 0, m = 1, \ldots, M$$

$$x \in X^0 = [\underline{x^0}, \overline{x^0}] = \{x \mid 0 < \underline{x_i^0} \leq x_i \leq \overline{x_i^0}, i = 1, \ldots, N\} \subset R^N$$

$$n_j(x) = \sum_{t=1}^{T1_j} \delta_{jt}^1 c_{jt}^1 \prod_{i=1}^{N} x_i^{y_{jti}^1}, j = 1, \ldots, p$$

$$d_j(x) = \sum_{t=1}^{T2_j} \delta_{jt}^2 c_{jt}^2 \prod_{i=1}^{N} x_i^{y_{jti}^2}, j = 1, \ldots, p$$

$$h_m(x) = \sum_{t=1}^{T3_m} \delta_{mt}^3 c_{mt}^3 \prod_{i=1}^{N} x_i^{y_{mti}^3}, m = 1, \ldots, M$$

Where $p \geq 2$ and all the numerators and denominators are generalized polynomial functions. The main idea behind this method is to find a lower bound for the solution of problem above and its sub-problems that are solved by linear programming. To this end, the authors proposed an exponential change of variable as $x_i = \exp(y_i)$ and then reformulated the problem as follows:

$$\min \quad g_0(y) = \sum_{j=1}^{p} \frac{n_j(y)}{d_j(y)} \tag{2.52}$$

$$\text{s.t.} \quad g_m(y) \leq 0, m = 1, \ldots, M$$

$$y \in Y^0 = [\underline{y^0}, \overline{y^0}] = \{y \mid \ln(\underline{x_i}) < y_i \leq \ln(\overline{x_i}), i = 1, \ldots, N\} \subset R^N$$

$$n_j(y) = \sum_{t=1}^{T1_j} \delta_{jt}^1 c_{jt}^1 \exp(\sum_{i=1}^{N} \gamma_{jti}^1 y_i), \quad j = 1, \ldots, p$$

$$d_j(y) = \sum_{t=1}^{T2_j} \delta_{jt}^2 c_{jt}^2 \exp(\sum_{i=1}^{N} \gamma_{jti}^2 y_i), \quad j = 1, \ldots, p$$

$$g_m(y) = \sum_{t=1}^{T3_m} \delta_{mt}^3 c_{mt}^3 \exp(\sum_{i=1}^{N} \gamma_{mti}^3 y_i), \quad m = 1, \ldots, M$$

Thus, using a three-level relaxation method, they presented a method to solve the equivalent problem by means of a simple bisection rule for partitioning in the branching process. In order to verify the theoretical results, they also provided a comparative analysis through some numerical examples tested in the literature. Given the value of ε, their algorithm is able to attain finite ε

convergence to the global minimum through a successive solution of a series of linear relaxation programs.

Other branch-and-bound algorithms were presented in [35] and [36] for sum of affine ratios and sum of quadratic ratios problems, respectively. In [35], maximization of sum of affine ratios is considered as below and an equivalent non-convex problem is proposed as follows:

$$\upsilon = \max h(x) = \sum_{i=1}^{p} \frac{n_i(x)}{d_i(x)} \tag{2.53}$$

$$\text{s.t.} \quad x \in X$$

$$\upsilon = \max \sum_{i=1}^{p} \frac{t_i}{s_i} \tag{2.54}$$

$$\text{s.t.} \quad n_i(x) - t_i \geq 0, i = 1, \ldots, p$$

$$-d_i(x) + s_i \geq 0, i = 1, \ldots, p$$

$$x \in X, l_i \leq t_i \leq u_i, L_i \leq s_i \leq U_i, i = 1, \ldots, p$$

Then using a convex relaxation program, the upper bounds of the problem are computed.

In [36], authors focused their attention on optimizing the quadratic sum of ratios using a new branch-and-bound technique. The problem is of the following form:

$$\min \quad H_0(y) = \sum_{i=1}^{p} \frac{f_i(y)}{g_i(y)} \tag{2.55}$$

$$\text{s.t.} \quad H_m(y) \leq 0, m = 1, \ldots, M$$

$$y \in Y^0=\{y \in R^n: \underline{y^0} \le y \le \overline{y^0}\} \subset R^n$$

where $p \geq 2$ and $f_i(y)$, $g_i(y)$ and $H_m(y)$ are all quadratic functions. A new parametric linearization method is developed to provide lower bounds of (2.55) and the sub-problems using an accelerated branch-and-bound algorithm that does not introduce new variables to the problem and, therefore, is more efficient compared to the previous methods used for the quadratic sums.

A more recent and novel approach in solving the sum of the ratios problem is presented by V. Gruzdeva and S. Strekalovsky in [37]. With an approach based on Dinkelbach's, the authors addressed the development of an efficient global search method to solve the multiple ratio fractional programs. They considered the two following problems:

$$\min \; f(x) = \sum_{i=1}^{m} \frac{\psi_i(x)}{\phi_i(x)} \tag{2.56}$$

$$\text{s.t.} \quad x \in S$$

$$\min \; \Phi(x,\alpha) = \sum_{i=1}^{m} [\, \psi_i(x) - \alpha_i \phi_i(x)] \tag{2.57}$$

$$\text{s.t.} \quad x \in S$$

wherein $\alpha = (\alpha_1, \dots \alpha_m)^T$, is a vector of parameters. They also introduced the function $\nu(\alpha)$ of the optimal value to the problem (1.64) as follows:

$$\nu(\alpha) = \inf\{\Phi(x,\alpha) \mid x \in S\} = \inf\{\sum_{i=1}^{m} [\, \psi_i(x) - \alpha_i \phi_i(x)]: x \in S\} \tag{2.58}$$

Furthermore, the following assumptions were made:

- $\nu(\alpha) > -\infty$ ∀α ∈ κ where κ is a convex compact set of R^m
- ∀α ∈ κ ∈ R^m there exists a solution $z = z(\alpha)$ to the problem (2.57)

Then they proved that any solution *z* to the problem (2.53) is also a solution to the original problem (2.56). The key contribution of this paper is the proposed search method to find an optimal set of values for the parameter α. To that end, they presented a global search technique to find α and then solve the parametric program. Thus, the algorithm consists of these steps: local and global searches in problem (2.57) with a fixed parameter α and the method for finding the vector parameter at which the optimal value of problem (2.58) is zero, i.e., $\nu(\alpha) = 0$. The authors tested their method on both affine and the quadratic sum of ratios, and the method proved to be efficient. However, the search method presented for finding the values of parameter α is shown to be inefficient and time-consuming due to a large number of iterations.

Kaiming Shen and Wei Yu [38] addressed the sum of ratios problem from a new perspective, proposing a novel transform to the original problem. This method does a similar job as the Dinkelbach's in terms of decoupling the numerator and the denominator, but unlike the latter, it can also be readily extended to the case of multiple ratios. The general form of the sum of ratios problem is given by:

$$\max \sum_{m=1}^{M} \frac{A_m(x)}{B_m(x)} \qquad (2.59)$$

$$\text{s.t.} \quad x \in X$$

And the authors proposed the following quadratic transform as an equivalent objective function:

$$\max \sum_{m=1}^{M} (2y_m\sqrt{A_m(x)} - y_m^2 B_m(x)) \tag{2.60}$$

$$\text{s.t.} \quad x \in X, y_m \in R$$

$$y_m^* = \frac{\sqrt{A_m(x)}}{B_m(x)}, m = 1, \ldots., M$$

They also generalized their transform to the case of the sum of functions of ratios and multi-objective fractional programs. The authors provided a comparative convergence analysis for the single ratio cases solved by Dinkelbach's algorithm and their proposed method and concluded that the novel quadratic transform can have slower convergence. They also applied their new transform to a group of multiple ratios problems in communications, including power control, beamforming, and energy efficiency in wireless networks, and proved the superiority of their approach to existing methods. They also provided more examples of discrete nature in their second paper [39].

Chapter Three: The New Variant of Dinkelbach's Algorithm for Sum of Ratios Optimization

In the chapter, the relationship between the multiple ratio fractional programs and parametric programming is introduced, and the algorithm to optimize a sum of ratios with concave numerators and convex denominators is presented. The mathematical formulation of the proposed algorithm is presented along with the results from numerical examples at the end of this chapter.

3.1 The relationship between multiple-ratio fractional programs and parametric programming

Let E^n be the Euclidean space of size n and S be a connected subset of it. Let $A_i(x)$ and $B_i(x)$ be set of continuous and real-valued functions of $x \epsilon S$. Also, it is assumed that all the $B_i(x)$ functions are positive, i.e., $B_i(x) > 0$ for all i and $x \epsilon S$.

The original problem is

$$\text{(I)} \qquad \max \textstyle\sum_i \frac{A_i(x)}{B_i(x)}$$

The proposed equivalent problem is

$$\text{(II)} \qquad \max \textstyle\sum_i A_i(x) - q\, B_i(x)$$

$$\text{For } q \epsilon E^1.$$

First, we need to prove that the proposed equivalent objective function is convex and therefore can be readily solved by means of any convex programming method. To this end, the following lemma is presented following the same approach as [6]:

Lemma 1: $F(q) = \max \sum_i A_i(x) - q\, B_i(x)$ is convex over E^1.

Proof: Let x_t maximise $F(tq' + (1-t)q'')$ with $q' \neq q''$ and $0 \leq t \leq 1$:

$F(tq' + (1-t)q'') = \sum_i A_i(x_t) - (tq' + (1-t)q'')B_i(x_t) =$

$\sum_i A_i(x_t) - (tq' + (1-t)q'')B_i(x_t) + tA_i(x_t) - tA_i(x_t) =$

$\sum_i t.\,(A_i(x_t) - q'B_i(x_t)) + (1\text{-}t).\,(A_i(x_t) - q''B_i(x_t)) \leq$

$t.\max \sum_i(A_i(x) - q'B_i(x)) + (1-t).\max \sum_i(A_i(x) - q''B_i(x)) =$

$tF(q') + (1-t)F(q'')$

Lemma 2: F(q) is continuous for $q \epsilon E^1$. (Proof in [40]).

Then we need to prove that the new objective function is strictly monotonic decreasing as follows:

Lemma 3: $F(q) = \max \sum_i A_i(x) - q\, B_i(x)$ is strictly monotonic decreasing, i.e., $F(q'') < F(q')$ if $q'' > q'$.

Proof: Let x''maximize $F(q'')$ then:

$F(q'') = \max \sum_i A_i(x) - q''\, B_i(x) =$

$\sum_i A_i(x'') - q''B_i\,(x'') < \sum_i A_i(x'') - q'B_i\,(x'') \leq \max \sum_i A_i(x) - q'\, B_i(x) = F(q')$

$\rightarrow\ F(q'') \leq\ F(q')$

Lemma 4: $F(q) = 0$ has a unique solution, say q_0.

Proof: This can be concluded from Lemma 2 and Lemma 3 and the following facts:

$$\lim_{q \to -\infty} F(q) = +\infty$$

$$\lim_{q \to +\infty} F(q) = -\infty$$

Next, we propose a formula for the auxiliary parameter q such that the value of F is always nonnegative based on the following lemma:

Lemma 5*:* Let $x^{+} \epsilon$ S and $q^{+} = \frac{\sum_i A_i(x)}{\sum_i B_i(x)}$ then $F(q^{+}) \geq 0$.

Proof: $F(q^{+}) = \max \sum_i A_i(x) - q^{+} B_i(x) \geq \sum_i A_i(x^{+}) - q^{+} B_i(x^{+}) = 0$

$\rightarrow F(q^{+}) \geq 0$

Finally, the following theorem is proposed and proved that suggests the new algorithm to solve the sum of ratios optimization problem:

Theorem*:* $q_0 = \frac{\sum_i A_i(x_0)}{\sum_i B_i(x_0)} = \max \frac{\sum_i A_i(x)}{\sum_i B_i(x)}$ if and only if, $F(q_0) = \max \sum_i A_i(x) - q_0 B_i(x) = 0$

Proof:

a) Let x_0 be a solution to the problem (I):

$q_0 = \frac{\sum_i A_i(x_0)}{\sum_i B_i(x_0)} \geq \frac{\sum_i A_i(x)}{\sum_i B_i(x)}$ for all x ϵ S.

α: $\sum_i A_i(x) - q_0 B_i(x) \leq 0$.

β: $\sum_i A_i(x_0) - q_0 B_i(x_0) = 0$.

From α: $F(q_0) = \max \sum_i A_i(x) - q_0 B_i(x) = 0$.

From β, we see that the maximum is taken on at x_0. Therefore, the first part of the proof is finished.

b) Let x_0 be a solution of problem (II) such that $\sum_i A_i(x_0) - q_0 B_i(x_0) = 0$. The definition of (II) implies that

$$\sum_i A_i(x) - q_0 B_i(x) \leq \sum_i A_i(x_0) - q_0 B_i(x_0) = 0 \quad \text{for all } x \in S.$$

Thus,

α: $\sum_i A_i(x) - q_0 B_i(x) \leq 0.$

β: $\sum_i A_i(x_0) - q_0 B_i(x_0) = 0$.

From α, we have $q_0 \geq \frac{\sum_i A_i(x)}{\sum_i B_i(x)}$ for $all\ x \in S$, that is q_0 is the maximum of the problem (I). From β, we have $q_0 = \frac{\sum_i A_i(x_0)}{\sum_i B_i(x_0)}$ that is x_0 is a solution vector of (I). Note that the theorem is also valid if we replace 'max' with 'min'.

3.2 Method for Solving the sum of ratios problem with concave A(x) and convex B(x)

Denoting an optimal solution of problem (I) with x_0, we can formulate this problem as below:

$$\text{Find } x_m\text{, such that } q(x_m) - q(x_0) < \varepsilon \text{ for any given } \varepsilon > 0.$$

Since F(q) is continuous, we have a second formulation:

$$\text{Find } x_n \text{ and } q_n = \frac{\sum_i A_i(x_n)}{\sum_i B_i(x_n)} \text{ such that } F(q_n) - F(q_0) = F(q_n) < \delta \text{ for any given } \delta > 0.$$

We also assume that $F(0) = \max\{\sum_i A_i(x) \mid x \in S\} \geq 0$. The algorithm can be started by setting q =0 (A_1) or by any feasible $x = x_1 \in S$ with $q(x) \geq 0, (A_2)$.

(A_1) Set $q_2 = 0$ and go to (B) with k =2.

(A_2) Let $x_1 \epsilon$ S and $q_2 = \frac{\sum_i A_i(x_1)}{\sum_i B_i(x_1)}$, proceed to (B) with k =2.

(B) By means of any method of concave programming, solve the following problem:

$$F(q_k) = \max\{\sum_i A_i(x) - q_k B_i\,(x)\},$$

and denote any solution point by x_k

(B_1) If $F(q_k) < \delta$: Stop. If $F(q_k) > 0$, then $x_k = x_n$

If $F(q_k) = 0$, then $x_k = x_0$

(B_2) If $F(q_k) \geq \delta$: Evaluate $q_{k+1} = \frac{\sum_i A_i(x_k)}{\sum_i B_i(x_k)}$ and go to (B) replacing q_k by q_{k+1}.

Proof of Convergence:

a) First, we need to prove that $q_{k+1} > q_k$ for all k with $F(q_k) \geq \delta$. Lemma 5 implies that $F(q_k) > 0$. By definition, we have:

$\sum_i A_i(x_k) = q_{k+1} \sum_i B_i(x_k)$.

$\rightarrow F(q_k) = \sum_i A_i(x_k) - q_k B_i\,(x_k) = q_{k+1} \sum_i B_i(x_k) - q_k \sum_i B_i(x_k) > 0.$

Since

$B_i(x) > 0$ then: $q_{k+1} > q_k$.

b) Next, we must prove that $\lim_{k\to\infty} q_k = q(x_0) = q_0$. If this is not true, we must have: $\lim_{k\to\infty} q_k = q^* < q_0$. By construction, we have a sequence x_k^* with q_k^*, such that $\lim_{k\to\infty} F(q_k^*) = F(q^*) = 0$ (see (B_1) of the algorithm). Since F(q) is strictly monotonic decreasing, we obtain: $0 = F(q^*) > F(q_0) = 0$, which is a contradiction. Therefore, it follows that $\lim_{k\to\infty} F(q_k^*) = F(q_0)$, and then by lemma 2, we have $\lim_{k\to\infty} q_k = q_0$.

The proposed algorithm is summarized as follows:

Table 1: Iterative Algorithm for Sum of Ratios Optimization

Algorithm 1 Iterative algorithm for concave-convex sum of ratios optimization problem
Initialization: Initialize x. Reformulate the problem using the modified Dinkelbach's transform according to (II) **Repeat** 1) Update q by $q_n = \frac{\sum_i A_i(x_n)}{\sum_i B_i(x_n)}$. 2) Update x by solving the new equivalent objective over x with fixed q. **Until** convergence.

3.3 Numerical Results

In this section, the performance of the proposed method is compared with the previous methods mentioned in the literature through some numerical examples. Examples include both affine and non-linear cases and are obtained from previously solved cases in [22-32].

3.3.1 Affine Examples

1- The first example involves maximization of sum of two ratios as below:

$$\max \quad \frac{3x_1 + x_2 - 2x_3 + 0.8}{2x_1 - x_2 + x_3} + \frac{4x_1 - 2x_2 + x_3}{7x_1 + 3x_2 - x_3}$$

$$\text{s.t.} \quad -x_1 - x_2 + x_3 \geq -1$$

$$x_1 - x_2 + x_3 \geq 1$$

$$-12x_1 - 5x_2 - 12x_3 \geq -34.8$$

$$-12x_1 - 12x_2 - 7x_3 \geq -29.1$$

$$6x_1 - x_2 - x_3 \geq 4.1$$

$$x_1, x_2, x_3 \geq 0$$

Based on the work of [22], the optimal point is (1, 0, 0), and the optimal value is 2.4714. Our method was able to achieve the exact same result in 3 iterations, whereas in [22], 23 iterations are needed to find the optimum.

2- The second example is a minimization task of the sum of two affine ratios as below:

$$\min \quad \frac{x_1 + 3x_2 + 2}{4x_1 + x_2 + 3} + \frac{4x_1 + 3x_2 + 1}{x_1 + x_2 + 4}$$

$$\text{s.t.} \quad -(x_1 + x_2) \leq -1$$

$$x_1, x_2 \geq 0$$

The authors in [26] claim that the optimal point is (1,0) and the optimal value is 1.4285, and their branch-and-bound algorithm requires ten iterations to converge. Our algorithm converges to the optimal point (1,0) and the value of 1.4285 in 2 iterations that are less complex compared to the mentioned method in [26].

3- Another example from [26] that involves sum of four ratios to be minimized as described below.

$$\min \quad -\left(\frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50}+\frac{3x_1+4x_3+50}{4x_1+4x_2+5x_3+50}+\frac{x_1+2x_2+5x_3+50}{x_1+5x_2+5x_3+50}+\frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50}\right)$$

$$\text{s.t.} \quad 2x_1 + x_2 + 5x_3 \le 10$$

$$x_1 + 6x_2 + 3x_3 \le 10$$

$$5x_1 + 9x_2 + 2x_3 \le 10$$

$$9x_1 + 7x_2 + 3x_3 \le 10$$

$$x_1, x_2, x_3 \ge 0$$

The results based on a branch-and-bound algorithm are optimal point (1.07, 0, 0) and the optimal value of - 4.087 in 17 iterations. Our method was able to find the point: (1.11, 0, 0) and the value of - 4.0907 as a near optimum point in 3 iterations.

4- Another example for sum and subtraction of four ratios that is described as below:

$$\max \quad \left(\frac{3x_1+4x_2+50}{3x_1+5x_2+4x_3+50}-\frac{3x_1+5x_2+3x_3+50}{5x_1+5x_2+4x_3+50}-\frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50}-\frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50}\right)$$

$$\text{s.t.} \quad 6x_1 + 3x_2 + 3x_3 \le 10$$

$$10x_1 + 3x_2 + 8x_3 \leq 10$$

$$x_1, x_2, x_3 \geq 0$$

The authors in [25] found the values of (-1.8e-16, 3.33, 0) and -1.9 as the optimal point and value respectively in 8 iterations. Using our proposed method, the optimal point (0, 3.3333, 0) and the value – 1.9 are obtained in 2 iterations.

5- For the following example, there were two different benchmarks for comparison. The first one is the results obtained from [28] that are (0, 0.625, 1.875) and 3.83, respectively. Another solution is found in [32] as follows: (0, 1.66, 0) and 3.71 as the min value, which is a lower minimum in 5 iterations. Our algorithm was able to find the exact results of [32] in only two iterations outperforming the two.

$$\min \ \left(\frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50} + \frac{3x_1+4x_3+50}{4x_1+4x_2+5x_3+50} + \frac{x_1+2x_2+4x_3+50}{x_1+5x_2+5x_3+50} + \frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50}\right)$$

$$\text{s.t.} \ \ 2x_1 + x_2 + 5x_3 \leq 10$$

$$x_1 + 6x_2 + 2x_3 \leq 10$$

$$9x_1 + 7x_2 + 3x_3 \geq 10$$

$$x_1, x_2, x_3 \geq 0$$

6- Another sum of affine ratios problem from [28] was tested, showing the superiority of our proposed method. The results obtained in [28] for the example below were: (0, 3.3333, 0) and the value of 3.0029. Although the algorithm converges early, the results can be shown to be infeasible according to the given constraints (the second constraint is violated).

Authors in [32] have found another solution for this example as: (5, 0, 0) and value of 2.8619 in 12 iterations. Using our method, the solution of (5, 0, 0) and value of 2.8619 were obtained in only three iterations which outperforms the method used in [32].

$$\min \quad \frac{3x_1 + 5x_2 + 3x_3 + 50}{3x_1 + 4x_2 + 5x_3 + 50} + \frac{3x_1 + 4x_2 + 50}{4x_1 + 3x_2 + 2x_3 + 50} + \frac{4x_1 + 2x_2 + 4x_3 + 50}{5x_1 + 4x_2 + 3x_3 + 50}$$

$$\text{s.t.} \quad 2x_1 + x_2 + 5x_3 \leq 10$$

$$x_1 + 6x_2 + 2x_3 \leq 10$$

$$9x_1 + 7x_2 + 3x_3 \geq 10$$

$$x_1, x_2, x_3 \geq 0$$

3.3.2 Non-Linear Examples

In order to evaluate the performance of our proposed algorithm, we need to test it with non-linear examples too. Therefore, the following examples are considered for this task.

1- The following example is considered as the first non-linear example to evaluate the performance of the proposed algorithm.

$$\max \quad \frac{x_1}{x_1^2 + x_2^2 + 1} + \frac{x_2}{x_1 + x_2 + 1}$$

$$\text{s.t.} \quad x_1 + x_2 \leq 1$$

$$x_1 \geq 0, x_2 \geq 0$$

Based on [31], the optimal value is 0.5958, and it is obtained in 7 iterations.

Our algorithm finds the near optimal value of 0.5833 and the point of (0.5, 0.5) in 3 iterations.

2- The second non-linear example is described as below:

$$\max \quad \frac{x_1}{x_1^2 + 1} + \frac{x_2}{x_2 + 1}$$

$$\text{s.t.} \quad x_1 + x_2 \leq 1$$

$$x_1 \geq 0, x_2 \geq 0$$

Again, the optimal point based on [31] is (0.5, 0.5) and value is 0.8. The results are obtained in 14 iterations. The same results are obtained in 3 iterations with our algorithm.

3- In this example, the number of variables is increased to three and the sum of two ratios need to be minimized as follows:

$$\min \quad \frac{x_1^2 - 4x_1 + 2x_2^2 - 8x_2 + 3x_3^2 - 12x_3 - 56}{x_1^2 - 2x_1 + x_2^2 - 2x_2 + x_3 + 20} + \frac{2x_1^2 - 16x_1 + x_2^2 - 8x_2 - 2}{2x_1 + 4x_2 + 6x_3}$$

$$\text{s.t.} \quad x_1 + x_2 + x_3 \leq 10$$

$$-x_1 - x_2 + x_3 \leq 4$$

$$x_1, x_2, x_3 \geq 1$$

The optimal point is (1.82, 1, 1) and value is – 6.1198 based on [31] in 6 iterations.

Using our algorithm, the near optimal point is (1.61, 1, 1) and value is – 6.1012 in 3 iterations.

4- Example 4 is defined as below in [31]:

$$\max \quad \frac{-x_1^2 + 3x_1 - x_2^2 + 3x_2 + 3.5}{x_1 + 1} + \frac{x_2}{x_1^2 - 2x_1 + x_2^2 - 8x_2 + 20}$$

$$\text{s.t.} \quad 2x_1 + x_2 \leq 6$$

$$3x_1 + x_2 \leq 8$$

$$x_1 - x_2 \leq 1$$

$$x_1, x_2 \geq 1$$

The optimal point is (1, 1.74), and the value is 4.06 in 9 iterations based on [31].

Using our algorithm, the near optimal point is (1.01, 1.75), and the value is 4.03 in 3 iterations.

5- Example 5 is described as below:

$$\min \ \frac{-x_1^2 + 3x_1 - x_2^2 + 3x_2 + 3.5}{x_1 + 1} + \frac{x_2}{x_1^2 + 2x_1 + x_2^2 - 8x_2 + 20}$$

$$2x_1 + x_2 \leq 6$$

$$3x_1 + x_2 \leq 8$$

$$x_1 - x_2 \leq 1$$

$$x_1 \geq 0.1, x_2 \leq 3$$

The optimal point is (0.1, 0.1) and value is 3.71 based on [31] and in 5 iterations. The same results are obtained using our algorithm in 2 iterations.

Based on the results obtained from running the algorithm for the examples above, our proposed method shows superior performance in terms of faster convergence in tested problems (with up to four ratios). In addition, this new method does a better job optimizing the objective function and finding a near optimum point than those mentioned in the references.

Chapter Four: Sum of Functions of Ratios Optimization in Wireless Networks

In previous chapters, the classic problems of fractional programming and different approaches to solving them were explored, and a new iterative method to optimize a particular category of these problems was developed and discussed. Now that we have a tool to solve this important group of problems, we need to see how this method performs in practice for more complicated problems like those found in communications. One of the most important examples of the sum of ratios optimization is found in wireless networks, where it is necessary to optimize sum energy efficiency or, in a more general expression, sum rate for a certain utility in the network in order to optimize the energy consumption in wireless networks[38]. For this purpose, this chapter is focused on the mentioned applications and is organized as follows: In the first section, the general problem of sum of functions of ratios is introduced and investigated, which models a broad category of problems found in the related communications literature. Then, different scenarios and problems are derived and discussed based on the choice of the function, and at the end, the numerical results of different scenarios are presented and compared to the literature.

4.1 Sum of Functions of Ratios Optimization

Let E^n be the Euclidean space of size n, and let S be a compact and connected subset of it. Let $A_i(x)$ and $B_i(x)$ be set of continuous and real-valued functions of $x \epsilon S$. Also, it is assumed that all the $B_i(x)$ functions are positive, i.e., $B_i(x) > 0$ for all i, and that $x \epsilon S$ and f is a non-decreasing function:

The general problem of sum of functions of ratios is defined as below:

$$\text{(III)} \quad \max \sum_i f\left(\frac{A_i(x)}{B_i(x)}\right).$$

The proposed equivalent problem is

$$\text{(IV)} \quad \max \sum_i f(A_i(x)) - q \times f(B_i(x)).$$

$$\text{For } q \epsilon E^1.$$

For this general case, we follow the same approach as the previous chapter to develop and prove the proposed theorem. We start by proving the convexity of the new formulation in as follows:

Lemma 1: $F(q) = \max \sum_i f(A_i(x)) - q \times f(B_i(x))$ is convex over E^1.

Proof: Let x_t maximise $F(tq' + (1-t)q'')$ with $q' \neq q''$ and $0 \leq t \leq 1$:

$F(tq' + (1-t)q'') = \sum_i f(A_i(x_t)) - (tq' + (1-t)q'') \times f(B_i(x_t)) =$

$\sum_i f(A_i(x_t)) - (tq' + (1-t)q'') \times f(B_i(x_t)) + tf(A_i(x_t)) - tf(A_i(x_t)) =$

$\sum_i t \times (f(A_i(x_t)) - q'f(B_i(x_t))) + (1\text{-}t)\times (f(A_i(x_t)) - q''f(B_i(x_t))) \leq$

$t \times \max \sum_i(f(A_i(x)) - q'f(B_i(x))) + (1-t) \times \max \sum_i(f(A_i(x)) - q''f(B_i(x))) =$

$tF(q') + (1-t)F(q'')$

Lemma 2*:* F(q) is continuous for $q\epsilon E^1$. (Proof can be easily deduced based on [40]).

Then we need to prove that the new objective function is strictly monotonic decreasing as follows:

Lemma 3*:* $F(q) = \max \sum_i f(A_i(x)) - q \times f(B_i(x))$ is strictly monotonic decreasing, i.e., $F(q'') < F(q')$ if $q'' > q'$.

Proof: Let x'' maximize $F(q'')$ then:

$F(q'') = \max \sum_i f(A_i(x)) - q'' \times f(B_i(x)) =$

$\sum_i f(A_i(x'')) - q'' \times f(B_i(x'')) < \sum_i f(A_i(x'')) - q' \times f(B_i(x'')) \leq \max \sum_i f(A_i(x)) - q' \times f(B_i(x)) = F(q')$.

$\rightarrow F(q'') \leq F(q')$.

***Lemma 4**:* $F(q) = 0$ has a unique solution, say q_0.

Proof: This can be concluded from Lemma 2 and Lemma 3 and the following facts:

$\lim_{q \to -\infty} F(q) = +\infty$

$\lim_{q \to +\infty} F(q) = -\infty$

Next, we propose a formula for the auxiliary parameter q such that the value of F is always nonnegative based on the following lemma:

***Lemma 5**: Let* $x^+ \epsilon S$ *and* $q^+ = \frac{\sum_i f(A_i(x))}{\sum_i f(B_i(x))}$ *then* $F(q^+) \geq 0$.

Proof: $F(q^+) = max \sum_i f(A_i(x)) - q^+ \times f(B_i(x)) \geq \sum_i f(A_i(x^+)) - q^+ \times f(B_i(x^+)) = 0$

$\rightarrow F(q^+) \geq 0$.

***Theorem**:* $q_0 = \frac{\sum_i f(A_i(x_0))}{\sum_i f(B_i(x_0))} = \max \frac{\sum_i f(A_i(x))}{\sum_i f(B_i(x))}$ if and only if, $F(q_0) = F(q_0, x_0) = \max \sum_i f(A_i(x)) - q_0 \times f(B_i(x)) = 0$

Proof:

c) Let x_0 be a solution of problem (I):

$$q_0 = \frac{\sum_i f(A_i(x_0))}{\sum_i f(B_i(x_0))} \geq \frac{\sum_i f(A_i(x))}{\sum_i f(B_i(x))} \quad \text{for all } x \in S$$

α: $$\sum_i f(A_i(x)) - q_0 \times f(B_i\,(x)) \leq 0$$

β: $$\sum_i f(A_i(x_0)) - q_0 \times f(B_i\,(x_0)) = 0$$

From α $F(q_0) = \max \sum_i f(A_i(x)) - q_0 \times f(B_i\,(x)) = 0$.

From β we see that the maximum is taken on at x_0. Therefore, the first part of the proof is finished.

d) Let x_0 be a solution of problem (II) such that $\sum_i f(A_i(x_0)) - q_0 \times f(B_i\,(x_0)) = 0$. The definition of (II) implies that

$$\sum_i f(A_i(x)) - q_0 \times f(B_i\,(x)) \leq \sum_i f(A_i(x_0)) - q_0 \times f(B_i\,(x_0)) = 0 \quad \text{for all } x \in S.$$

Thus,

α: $$\sum_i f(A_i(x)) - q_0 \times f(B_i\,(x)) \leq 0.$$

β: $$\sum_i f(A_i(x_0)) - q_0 \times f(B_i\,(x_0)) = 0.$$

From α, we have $q_0 \geq \frac{\sum_i f(A_i(x))}{\sum_i f(B_i(x))}$ for all $x \in S$, that is q_0 is the maximum of the problem (I). From β we have $q_0 = \frac{\sum_i f(A_i(x_0))}{\sum_i f(B_i(x_0))}$ that is x_0 is a solution vector of (I). Note that the theorem is also valid if we replace 'max' with 'min'.

4.2 Method for Solving the sum of functions of ratios problem with concave A(x) and convex B(x)

Denoting an optimal solution of problem (I) with x_0, we can formulate this problem as follows:

$$\text{Find } x_m, \text{ such that } q(x_m) - q(x_0) < \varepsilon \text{ for any given } \varepsilon > 0.$$

Since F(q) is continuous, we have a second formulation:

$$\text{Find } x_n \text{ and } q_n = \frac{\sum_i f(A_i(x_n))}{\sum_i f(B_i(x_n))} \text{ such that } F(q_n) - F(q_0) = F(q_n) < \delta \text{ for any given } \delta > 0.$$

We also assume that $F(0) = \max\{\sum_i f(A_i(x)) \mid x \in S\} \geq 0$. The algorithm is given as follows, with one of two starting points: either q =0 (A_1) or by any feasible $x = x_1 \in S$ with $q(x) \geq 0$, (A_2).

(A_1) Set $q_2 = 0$ and go to (B) with k =2.

(A_2) Let $x_1 \in S$ and $q_2 = \frac{\sum_i f(A_i(x_1))}{\sum_i f(B_i(x_1))}$, proceed to (B) with k =2.

(B) By means of any method of concave programming, solve the following problem:

$$F(q_k) = \max\{\sum_i f(A_i(x)) - q_k \times f(B_i(x))\}.$$

And denote any solution point by x_k

(B_1) If $F(q_k) < \delta$: Stop. If $F(q_k) > 0$, then $x_k = x_n$

If $F(q_k) = 0$, then $x_k = x_0$

(B_2) If $F(q_k) \geq \delta$: Evaluate $q_{k+1} = \frac{\sum_i f(A_i(x_k))}{\sum_i f(B_i(x_k))}$ and go to (B) replacing q_k by q_{k+1}.

Proof of Convergence:

a) First, we need to prove that $q_{k+1} > q_k$ for all k with $F(q_k) \geq \delta$. Lemma 5 implies that $F(q_k) > 0$. By definition, we have:

$\sum_i f(A_i(x_k)) = q_{k+1} \sum_i f(B_i(x_k))$.

$\rightarrow F(q_k) = \sum_i f(A_i(x_k)) - q_k \times f(B_i(x_k)) = q_{k+1} \sum_i f(B_i(x_k)) - q_k \times \sum_i f(B_i(x_k)) > 0$.

Since

$B_i(x) > 0$ and f is non-decreasing, then: $q_{k+1} > q_k$.

b) Next, we must prove that $\lim_{k\to\infty} q_k = q(x_0) = q_0$. If this is not true, we must have: $\lim_{k\to\infty} q_k = q^* < q_0$. By construction, we have a sequence x_k^* with q_k^*, such that $\lim_{k\to\infty} F(q_k^*) = F(q^*) = 0$ (see (B_1) of the algorithm). Since F(q) is strictly monotonic decreasing, we obtain: $0 = F(q^*) > F(q_0) = 0$, which is a contradiction. Therefore, it follows that $\lim_{k\to\infty} F(q_k^*) = F(q_0)$, and then by lemma two, we have $\lim_{k\to\infty} q_k = q_0$.

The algorithm is summarized as follows:

Table 2: Iterative Algorithm for Sum of Functions of Ratios Optimization

Algorithm 2 Iterative algorithm for concave-convex sum of functions of ratios optimization problem
Initialization: Initialize x. Reformulate the problem using the modified Dinkelbach's transform according to (IV) **Repeat** 1) Update q by $q_n = \frac{\sum_i f(A_i(x_n))}{\sum_i f(B_i(x_n))}$.

2) Update x by solving the new equivalent objective over x with fixed q.

Until convergence.

4.3 Fractional Programming in Wireless Communications

Optimization is an important part of any communication systems design [41], [42]. So far, we have explored the fundamentals of Fractional Programming and developed a new variant of Dinkelbach's method to solve the general problem of the sum of functions of ratios. The rest of this document is focused on applications of Fractional Programming in communication systems.

According to [43], the Information and Communication Technologies (I.C.T.) community is contributing 5% to the global footprint of CO_2 emissions. This might seem a small amount considering the rapid development of new technologies such as the Internet of Things and connected devices, the demand for energy is constantly increasing [44]. Adjusting to this new demand can not easily be performed by increasing the transmit power due to challenges like sustainable development and economic growth. Thus, we need to incorporate more efficient technologies into the communication systems infrastructures aiming to optimize the use of energy to get higher data rates. This leads us to the very well-known concept of Energy-Efficient Resource Allocation in wireless networks, which is related to the key performance indicators in 5G networks.

Energy-efficient metrics are often modeled by fractional functions, and therefore, using fractional programming techniques seems to be the appropriate way to approach these problems. In [45] the most recent applications of fractional programming in communication systems were studied, including Multi-Carrier Communications, Multiple Antenna Communications, Relay-Assisted Communications, Device-to-Device, and Cognitive Radio and 5G Networks. The authors also emphasized the fact that in 5G networks, where the presence of interference can not be avoided, the use of interference mitigation techniques is strongly recommended in order to be able to use fractional programming methods to optimize the energy efficiency metrics. This is due to the presence of interference terms in the energy efficiency formula which creates non-concave numerators in the fractional terms. The mathematical logic behind this problem is presented in more detail after the definition of energy efficiency is given in the following section.

4.3.1 Energy Efficiency Optimization in Wireless Networks

The definition of efficiency in a system that uses a certain resource is described as the ratio between the produced service and the cost [45]. In the wireless communication context, the cost is defined as the amount of used energy to establish the communication link. If the transmit power is denoted by p and the transmission time is T, the consumed energy is written as follows:

$$\mathrm{E(p) = T(p + P_c)\ [Joule].} \tag{4.1}$$

wherein P_c is the circuit power. In most practical cases [46], the transmit amplifier is considered to work in the linear region, and the value of circuit power (P_c) is fixed, and it does not depend on the power budget. The system benefit can be defined as the amount of data that a link can communicate reliably. One way to measure this is through system capacity in communication systems. The capacity of a wireless network in an interval T is measured as $Tf(\gamma) = T.B.(log(1+\gamma))$ with B denoting the bandwidth and γ the signal-to-noise + interference (SINR) ratio. Based on the above assumptions, the energy efficiency (E.E.) of a wireless link is defined as:

$$\mathrm{EE} = \frac{\mathrm{Tf}(\gamma)}{\mathrm{T}(\mathrm{p}+\mathrm{P_c})} = \frac{\mathrm{f}(\gamma)}{(\mathrm{p}+\mathrm{P_c})}\ [\mathrm{bits/Joule}]. \tag{4.2}$$

Where p is the power assigned to each user and P_c is the circuit power.

To develop a more general model for wireless networks, we consider a network with K transmitters and M receivers with a bandwidth of B. We define $\boldsymbol{p} = \{p_k |\ k = 1,2, \dots K\}$ with p_k the *k*th user's transmit power. Therefore, the E.E. for link *k* can be written as:

$$\mathrm{EE_k} = \frac{\mathrm{f_k(p)}}{\mathrm{p_k}+\mathrm{P_{c,k}}}. \tag{4.3}$$

wherein $f_k(p)$ is the SINR for the user *k*. As for the network-wide measure of energy efficiency, we have different definitions, and amongst those, the Global Energy Efficiency (G.E.E.) and Weighted Sum Energy Efficiency (WSEE) are the most well-established metrics. Since the metric used in our model is WSEE, we only mention the definition of WSEE as below:

$$\mathrm{WSEE} = \sum_{\mathrm{k}=1}^{2\mathrm{K}} \mathrm{w_k EE_k}. \tag{4.4}$$

The weights w_k are set to satisfy the E.E. priorities of different users.

A massive MIMO system with K single-antenna user pairs is considered. The users are connected through a relay that amplifies and forwards the signal to the receivers. Thus, there is no direct link between the users, and the connection is established through the relay [47]. The network operates in two modes; first, the users send the signal to the relay, which is considered as Multiple Access mode, and then the relay broadcasts the signal to the other half of the users. Both the users and the relay operate in a time division duplex, and thus the Multiple Access and Broadcast channels are reciprocal. First, each user sends the signal $\sqrt{p_k}s_k$ to the relay wherein p_k denotes the power of user S_k. The received signal at the relay at this stage can be modeled as follows:

$$y_R = \sum_{k=1}^{2K} \sqrt{p_k} g_k s_k + n_R = \widetilde{G}s + z_R. \quad (4.5)$$

where $s = [s_1, s_2, \ldots, s_k]^T$, $G = GP$ with P = diag $\{\sqrt{p_1}, \sqrt{p_2}, \ldots, \sqrt{p_{2K}}\}$ and $G = [g_1, g_2, \ldots g_{2K}]$ is the channel matrix with g_k being the channel between the relay and the user *k*. The elements of G with $G_{kk} = \beta_k$, denote the fading coefficients. The relay noise vector z_R is modeled with $N(0, \sigma_{nr}^2)$.

After receiving the signal, the relay amplifies the received signal as $s_R = \alpha W y_R$ and then forwards it to the users. Here, α is the amplification coefficient, and W is the precoder matrix. The signal received by the user k' is:

$$y_{k'} = g_{k'}^T S_R + z_{k'} = \alpha g_{k'}^T W \sqrt{p_k} g_k s_k + \alpha g_{k'}^T W \sum_{i \neq k}^{2K} \sqrt{p_i} g_i s_i + \tilde{z}_{k'}. \quad (4.6)$$

The received signal on the other side of the relay includes both the desired signal and the interference. It is assumed that the interference terms are canceled out by the relay using precoding methods [47].

With energy efficiency defined previously as $EE_k = \frac{R_k}{p_k + P_{c,k}}$,we can define the weighted sum energy efficiency (WSEE) optimization problem as follows [48]:

$$\mathbf{P1}: \max_{p_k, P_R} \sum_{k=1}^{2K} w_k \frac{\log_2(1+SNR(p_k))}{p_k + P_{c,k}} \quad (4.7)$$

$$\text{s.t. } 0 \le p_k \le P^{max}, 0 \le P_R \le P_R^{max}$$

$$\sum_{k=1}^{2K} p_k + P_R \le P_t^{max}$$

$$R_k \ge \bar{R}_k.$$

Wherein P .R. denotes the relay power, and P_t^{max} indicates the maximum available transmit power. The constraints are defined such that each user has maximum assigned power, and the relay power has a limit too. Also, the minimum quality of service (QoS) for each user is determined as $\bar{R}_k$.

The formula we used for the SNR in a multi-pair two-way A.F. half-duplex massive MIMO relay with N "K and a precoder for a high SNR case is derived in [47] as below:

$$\mathrm{SNR}(p_k) = \frac{p_k(N-2K-1)\beta_k}{\sigma_{nr}^2}. \tag{4.8}$$

$$R_k = \log_2(1 + \mathrm{SNR}(p_k)). \tag{4.9}$$

As can be seen, the objective function in **P1** has a set of concave numerators and convex denominators, which satisfies the required conditions for the application of our new variant of Dinkelbach's algorithm. Therefore, the problem can be modeled under the general problem of sum of functions of ratios optimization with the non-decreasing function being $f(x) = x$, leading to the classic sum of ratios optimization problem discussed in the second chapter. Here, to solve the optimization problem, we numerically calculate WSEE using the new variant of Dinkelbach's algorithm, the Quadratic transform approach discussed in [38], and another similar approach based on bisection search in [37]. We consider a noise power of -20 dBm and P^{max} of 30 dBm, and P_t^{max} of 10 dBm for a scenario with five user pairs and N=64 antennas at the relay. Similar to [49], the large-scale fading matrix is chosen as:

$$\mathrm{G} = \mathrm{diag}[0.749, 0.045, 0.246, 0.121, 0.125, 0.142, 0.635, 0.256, 0.021, 0.123] \tag{4.10}$$

and the weights w_k that are chosen randomly in a way to satisfy the condition $\sum_{\mathrm{k}=1}^{2\mathrm{K}} \mathrm{w_k} = 1$, are chosen as:

$$\mathrm{w_k} = [0.15, 0.09, 0.06, 0.12, 0.19, 0.11, 0.08, 0.05, 0.12, 0.03]. \tag{4.11}$$

The performance of the proposed algorithm is compared to the other mentioned methods for a QoS requirement of at least 0.5 bps/Hz for each user and $P_R = P_t^{max}/2$. The WSEE is plotted versus the number of iterations in figure 1. We observe that both the proposed algorithm and the bisection search method yield the same value for WSEE, whereas the quadratic approach is only able to converge to a local maximum that is much lower than the maximum achieved with the two other methods. In terms of complexity, it should be noted that both the bisection search method and the quadratic method require to find and update ten auxiliary variables (equal to the number of ratios in the sum) for their proposed equivalent objective function, whereas for our proposed method, we only need to find and iteratively update one auxiliary parameter regardless of the number of ratios. This makes the proposed method superior in terms of faster iterations and simplicity.

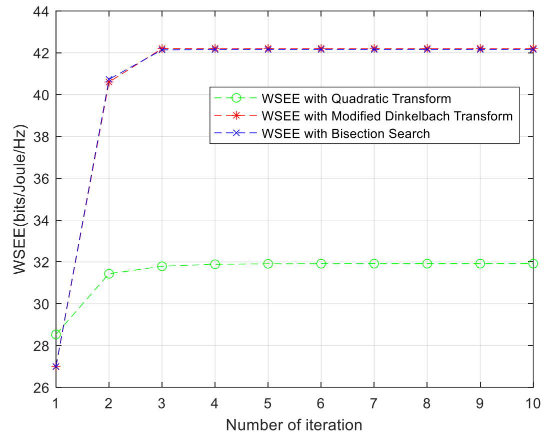

Figure 1: WSEE for different methods with $P_t^{max} = 10\ dBm$

The same scenario for the case of P_t^{max} = 20 dBm is repeated, and the WSEE is plotted for three different methods versus the number of iterations:

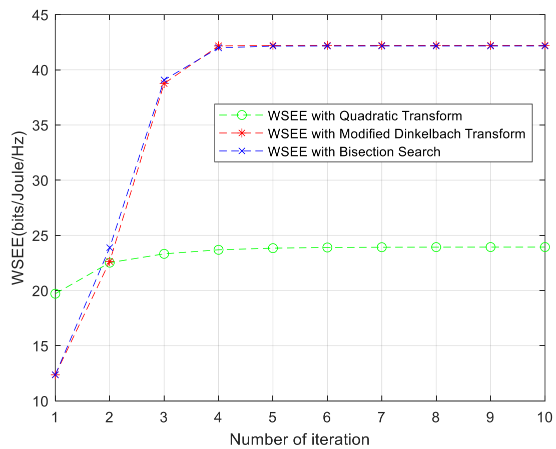

Figure 2: WSEE for different methods with $P_t^{max} = 20\ dBm$

Once again, both the proposed algorithm and the bisection search method seem to find the exact solution for the problem while the quadratic transform method only converges to a local maximum.

For the case of $P_t^{max} = 10$ dBm, the obtained value of WSEE versus the number of relay antennas is plotted in figure 3:

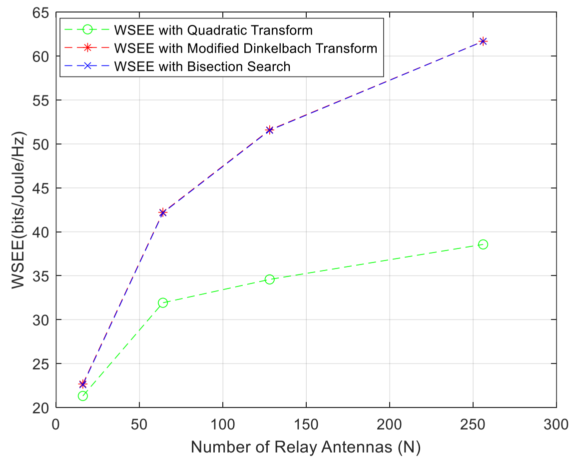

Figure 3: WSEE versus number of relay antennas

We observe that WSEE increases with increasing the number of antennas for both methods mainly because with the increase in N, the channel for all users becomes closer to orthogonal. As a result, the beam formed by the relay points to the target with more precision. The improvement in the obtained value is more evident for the Modified Dinkelbach's method, which confirms the superiority of this method in another way considering the simplicity and convergence speed of the algorithm. The Bisection search method shows a similar reaction to the change in the number of relay antennas, but it should be noted that each iteration in this algorithm takes much more time than the proposed method; thus, the most reliable and fast algorithm is the modified Dinkelbach's method. Also, both the Bisection search and the Quadratic transform methods require calculating and updating ten parameters (the number of ratios) while the proposed algorithm does the job only by introducing one auxiliary variable to the equivalent objective formula.

The obtained results for the modified Dinkelbach's transform are compared with the other algorithms in the following table:

Table 3: Computational Complexity of the Algorithms for WSEE Example

Problem	Number of ratios	Number of variables	Convergence value	Number of iterations	Time
Modified Dinkelbach's Transform	10	10	42.21	3	4.53 s
Quadratic Transform	10	10	31.9	5	5.21 s
Bi-Section Search	10	10	42.15	3	6.34 s

4.3.2 Power Control in Wireless Networks

Power control in interference-limited networks has always been a critical challenge in the design and implementation of the networks. A very common way to formulate this optimization problem is through weighted sum-rate maximization. Previous work in this area is mainly focused on a special case where SINR is much higher than 0 dB and the term $\log_2 w_i(1 + SINR)$ in the rate formula can be approximated by $\log_2 w_i(SINR)$ and then solved by means of Geometrical Programming. However, in most practical cases, this assumption does not hold [50]. In this section, we propose a method based on the algorithm for the sum of functions of ratios programming to maximize the weighted sum rate in a general SINR regime.

A multi-pair massive MIMO system is considered with M = 4 single antenna user pairs. The channel gain between the transmitters and receivers is denoted by $G_{i,j}$. Let p_i denote the transmission power of link i and n_i denote the link i. The received SINR of the link i can be written as:

$$\gamma_i(p) = \frac{G_{ii} p_i}{\sum_{j \neq i} G_{ji} p_j + n_i}. \tag{4.12}$$

The data rate is calculated based on the Shannon formula as $\log_2(1 + \gamma_i(p))$. The problem is defined as finding the optimal power allocation scheme in order to maximize the weighted sum rate objective function subject to the rate and power constraints. The mathematical formulation of the problem is as follows:

$$\textbf{P2:} \ \max \sum_{i=1}^{M} w_i \log_2\big(1 + \gamma_i(p)\big) \tag{4.13}$$

$$\text{s.t.} \ \log_2\left(1 + \gamma_i(p)\right) \geq r_{i,min}.$$

$$0 \leq p_i \leq p_i^{max}.$$

Here, $r_{i,min} \geq 0$ is the minimum data rate for the link i and $w_i > 0$ denotes the priority of link i. Note that the priority weights are normalized so that $\sum_{i=1}^{M} w_i = 1$.

As can be seen, the objective function in **P2** has the form of the sum of functions of ratios except for the value one inside the log function. As discussed before, there is a trivial solution to this problem in the case of high values of SINR using geometrical programming. In order to find a more accurate and general solution for this problem, we propose using the Taylor series approximation of the term $\log_2\big(1 + \gamma_i(p)\big)$ as below:

$$\log_2\big(1 + \gamma_i(p)\big) = \gamma_i(p) - \frac{\gamma_i(p)^2}{2} + \frac{\gamma_i(p)^3}{3} - \frac{\gamma_i(p)^4}{4} + \ldots,. \tag{4.14}$$

Considering only the first term of the above expression, we arrive at the following approximation for problem **P2**:

$$\textbf{P3:}\ \max \sum_{i=1}^{M} w_i\, \gamma_i(p) \tag{4.15}$$

$$\text{s.t.}\quad \log_2\left(1+\gamma_i(p)\right) \geq r_{i,\min}.$$

$$0 \leq p_i \leq p_i^{\max}.$$

The objective function of problem **P3** is comprised of concave numerators and convex denominators, which satisfies the required conditions for the application of the new variant of Dinkelbach's algorithm. Therefore, the problem can be modeled under the general problem of sum of functions of ratios optimization with the non-decreasing function being $f(x) = x$, leading to the classic sum of ratios optimization problem discussed in the second chapter. Here, to solve the optimization problem, we numerically calculate the weighted sum rate using the new variant of Dinkelbach's algorithm and the Quadratic transform approach discussed in [38]. We consider a four-link network with the channel gain matrix as follows [50] :

$$G = [0.4310\ \ 0.0002\ \ 0.2605\ \ 0.0039; \tag{4.16}$$

$$0.0002\ \ 0.3018\ \ 0.0008\ \ 0.0054;$$

$$0.0129\ \ 0.0005\ \ 0.4266\ \ 0.1007;$$

$$0.0011\ \ 0.0031\ \ 0.0099\ \ 0.0634].$$

$P^{\max} = [0.7, 0.8, 0.9, 1]$ and $n_i = 0.1\,\mu W$ for all links. The priority weights are chosen as $w = [1/6, 1/3, 1/6, 1/3]$. Also, the minimum rate constraint is not considered in this example. The value of the weighted sum rate versus the number of iterations is plotted in figure 4:

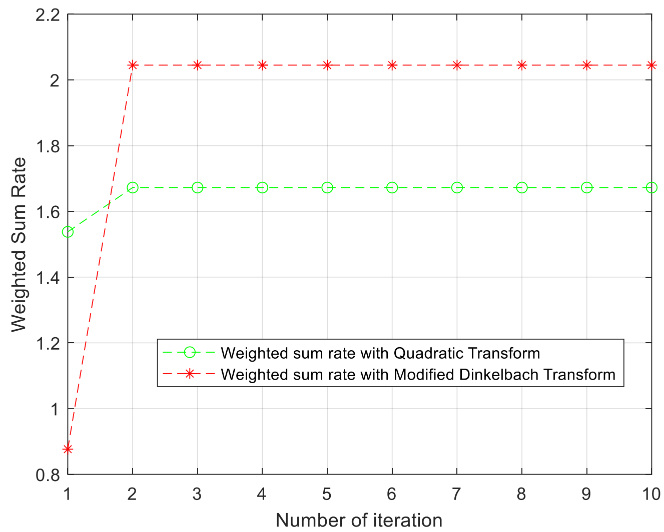

Figure 4: Weighted Sum Rate versus the number of iterations

We observe that the proposed algorithm does a better job finding a greater maximum value for the weighted sum rate optimization problem. On the other hand, the quadratic transform approach converges to a smaller value. It should be noted that the modified Dinkelbach's approach only needs to find and update one auxiliary variable in each iteration, whereas for the quadratic transform approach, the algorithm needs to find and update a parameter for each ratio in the sum. Thus, the superiority of the proposed approach in terms of accuracy and complexity is proved.

The computational complexity of our algorithm is compared with that of the Quadratic transform in the following table:

Table 4: Computational Complexity of the Algorithms for Power Control Example

Problem	Number of ratios	Number of variables	Convergence value	Number of iterations	Time
Modified Dinkelbach's Transform	4	4	2.04	2	1.12 s
Quadratic Transform	4	4	1.67	2	9.25 s

4.3.3 Beamforming Optimization in Wireless Networks

The problem of weighted sum-rate maximization in the downlink of a multicell network is considered. Many non-linear precoding schemes are proposed to achieve the required capacity that is considered of high complexity [51]. This problem, even for the case of single-antenna receivers, is known to be an NP-hard problem [52]. Previous work in this area [53] proposes to find the suboptimal beamformers using a Branch-and-Bound algorithm, but as the authors mention, the complexity of the solution grows exponentially with the problem size. In this section, we propose a solution to find the optimal beamformers in a downlink MISO system to maximize the weighted sum rate objective. The details of the considered scenario are presented below.

A system of B coordinated B.S.s with N transmit antenna, and K single-antenna receivers are considered. The set of the users is denoted by $U = \{1, 2, \ldots, K\}$ and the set of the B.S.s is shown with B = {1, 2, …, B}. Also, the set of the users served with B.S. b is denoted by U_b. The received signal in the flat fading regime is computed as below:

$$y_k = h_{b_k,k} w_k d_k + \sum_{i=1, i\neq k}^{K} h_{b_i,k} w_i d_i + n_k. \tag{4.17}$$

wherein $h_{b_i,k} \subset C^{1\times N}$ is the channel vector from B.S. b_i to user k and $w_k \subset C^{1\times N}$ is the beamforming vector from B.S. b_k to user k, and d_k is the normalized data symbol, and n_k is Gaussian noise with variance σ^2. It is assumed that the term $\sum_{i=1,i\neq k}^{K} h_{b_i,k} w_i d_i$ are modeling both the inter-cell and intra-cell interference [54].

The total power transmitted by BS b is $\sum_{k\epsilon U_b} ||w_k||_2^2$ and the SINR γ_k for the user k is denoted by

$$\gamma_k = \frac{|h_{b_k,k} w_k|^2}{\sigma^2 + \sum_{i=1,i\neq k}^{K} |h_{b_i,k} w_i|^2}. \tag{4.18}$$

The problem to be solved is defined as maximizing the weighted sum rate subject to per B.S. power constraints and is modeled as follows:

$$\textbf{P4:} \quad \max_{\sum_{k\epsilon U_b} ||w_k||_2^2 \leq P_b} \sum_{k=1}^{K} \alpha_k \log(1+\gamma_k). \tag{4.19}$$

where α_k is introduced to model the priority of each user as a positive coefficient, again, this problem is a weighted sum of functions of ratios and can be solved by our proposed method based on the new variant of Dinkelbach's algorithm. To solve the problem, we consider a single cell case where one B.S. with N = 4 antennas is serving K = 4 users. The channel is Gaussian, and the noise variance is assumed to be 0 dBm. The total transmit power at the BS is 20 dBm. Here, we assume that all the users have the same priority, and therefore, the value of α_k is 1 for all users. The achieved sum rate of our proposed method is compared with the Quadratic method in [38], and the results are plotted in figure 5:

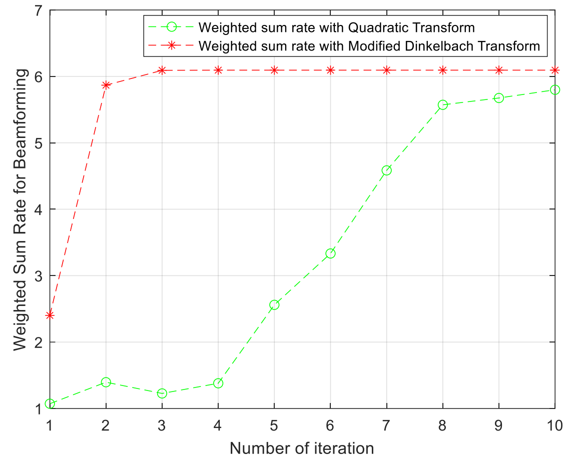

Figure 5: Obtained weighted sum rate versus number of iterations

It is observed that, as expected, the new variant of Dinkelbach's algorithm is converging to a stationary point in a few iterations while the other method is showing a much slower convergence and to a lower maximum. In addition, some fluctuations are seen throughout the course of running the algorithm.

The computational complexity of our algorithm is compared with that of the Quadratic transform in table below:

Table 5: Computational Complexity of the Algorithms for Beamforming Example

Problem	Number of ratios	Number of variables	Convergence value	Number of iterations	Time
Modified Dinkelbach's Transform	4	16	6.09	3	1.11 s
Quadratic Transform	4	16	5.79	10	16.07 s

The achieved sum rate versus the total transmit power at the B.S. is plotted in figure 6. It is observed that both algorithms yield better results when the total transmit power increases.

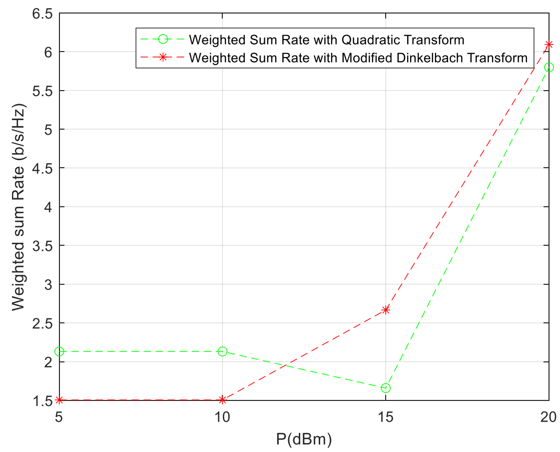

Figure 6: Obtained weighted sum rate versus total available power

Chapter Five: Discussion and Future Work

The general problem of the sum of functions of ratios optimization was studied in this thesis, and the relevant published work was presented in different scenarios. Using a modified version of Dinkelbach's algorithm, we proposed a fast and effective method for the optimization of the sum of non-decreasing functions of ratios. The key challenge about this group of optimization problems is that the general formulation is non-convex, and we need to find an alternative convex objective function that approximates the original problem.

Using the new variant of Dinkelbach's algorithm, we developed a framework to effectively tackle the well-known problem of sum-rate optimization with concave numerators and convex denominators. As discussed in the previous chapters, our method outperforms the previously discussed methods, including the proposed methods in [37] and [38] in terms of convergence speed and accuracy in finding a stationary point. The quadratic method proposed in [38] is more computationally complex than our algorithm since it requires finding and updating more parameters for the equivalent objective function. In other words, the number of auxiliary parameters needed for the algorithm based on the quadratic transform to define the equivalent objective function is equal to the number of ratios in the problem. Whereas, in the proposed new variant of Dinkelbach's algorithm, there is only one auxiliary parameter incorporated into the equivalent objective function regardless of the number of ratios. This makes the proposed method less computationally complex compared with competing algorithms. In addition, as the authors state in [38], the method based on quadratic transform finds a local optimum, and this fact is confirmed in the conducted simulations in chapter three while our proposed method does a better job in converging to a near optimal point.

The new variant of Dinkelbach’s method also outperforms the method in [37] in terms of both complexity and convergence speed. The algorithm in [37] is based on a bi-section search method to find the auxiliary parameters needed to define the equivalent objective function. In each iteration, we need to run a search algorithm to find and update the parameters, which can be a time-consuming process in many applications, especially with the increase in the number of ratios in the problem. However, the results are consistent with those obtained with our proposed method, and the only difference is in the amount of time that it takes for each algorithm to find and update the auxiliary variables.

The obtained results with the new variant of Dinkelbach’s algorithm could be applied in many applications in communication systems. One great impact of using this fast and accurate algorithm could be in optimizing the use of energy in a wireless network while ensuring a minimum level of quality of service for each user. Using the new variant of Dinkelbach’s algorithm, the general problem resource allocation in a communication link can be effectively addressed. As a result, we can have less energy consumption in wireless applications such as Internet of Things and other network-based services.

Given that most of the work in the sum of ratios optimization is based on Branch-and-Bound algorithms, these results provide new insights into the sum of ratios optimization problem by leveraging the simplicity of Dinkelbach’s algorithm and incorporating some changes to the original formulations. As previously said, the new formulation also includes the original Dinkelbach’s algorithm for the case of single ratio optimization. For many years, it was believed that no generalization to Dinkelbach's algorithm was possible, and a few attempts like [14] were proven wrong later [15].

Although the new variant of Dinkelbach's algorithm shows promising performance for the conducted numerical examples and in more practical scenarios of communications, it should be noted that generalizability of the results is limited to the cases with the concave numerator and convex denominator. This implies the fact that for a problem such as weighted sum energy efficiency, interference-limited scenarios could not be solved with this method due to the non-concave terms incorporated by the interference in the numerator. Also, the performance of the algorithm is only confirmed for up to ten ratios, and the results for bigger problems are unknown due to not having access to previously solved problems as a benchmark. Future studies should consider conducting simulations for much larger problems and evaluating the performance of the proposed algorithm for those cases. Additionally, the application of the proposed method in optimizing the sum of other non-decreasing functions of ratios such as the sum of squared ratios can be investigated in future work.

References

[1] Edwin K. P. Chong, Stanislaw H. Zak, An Introduction to Optimization, John Wiley & Sons, 2004.

[2] Von Neumann, J., "Uber ein ¨okonomisches Gleichungssystem und eine Verallgemeinerung des Brouwerschen Fixpuntsatzes," *Ergebnisse eines mathematischen Kolloquiums* , pp. 73-83, 1937.

[3] Abraham Charnes, William W Cooper, "Programming with linear fractional functionals," *Naval Research logistics quarterly,* vol. 9, no. (3-4), pp. 181-186, 1962.

[4] J.B.G. Frenk, S. Schaible, "Fractional Programming," ERIM Report Series, 2004.

[5] S. Schaible, "Parameter-free convex equivalent and dual programs of fractional programming problems," *Zeitschrift f¨ur Operations Research,* vol. 18, no. 5, pp. 187-196, 1974.

[6] W. Dinkelbach, "On Nonlinear Fractional Programming," *Management Science,* vol. 13, no. 7, pp. 492-498, 1967.

[7] R. W. Freund and F. Jarre, "Solving the sum-of-ratios problem by an interior-point method," *Journal of Global Optimization,* pp. 82-102, 2001.

[8] J. Shi, "A Combined Algorithm for Fractional Programming," *Annals of Operations Research,* vol. 103, pp. 135-147, 2001.

[9] H. P. Benson, "Fractional programming with convex quadratic forms and functions," *European Journal of Operational Research,* vol. 173, no. 2, pp. 351-369, 2006.

[10] Jun-Ya Gotoh, Hiroshi Konno, "Maximization of the Ratio of Two Convex Quadratic Functions over a Polytope," *Computational Optimization and Applications,* vol. 20, pp. 43-60, 2001.

[11] Sapan Kumar Das, T. Mandal, "A single-stage single constraints linear fractional programming problem: An approach," *Operations Research and Applications: An International Journal (ORAJ),* vol. 2, no. 1, 2015.

[12] Sumon Kumar Saha, Md. Rezwan Hossain, Md. Kutub Uddin, Rabindra Nath Mondal, "A New Approach of Solving Linear Fractional Programming Problem by using Computer Program," *Open Journal of Optimization,* vol. 4, pp. 74-86, 2015.

[13] Siegfried Schaible, Jianming Shi, "Fractional programming: The sum-of-ratios case," *Optimization Methods and Software,* vol. 18, no. 2, 2003.

[14] Y. Almogy, O. Levin, "A class of fractional programming problems," *Operations Research,* vol. 19, pp. 57-67, 1971.

[15] J.E. Falk and S.W. Palocsay, "Optimizing the sum of linear fractional functions," *Recent Advances in Global Optimization,* pp. 221-258, 1992.

[16] Jean-Pierre Crouzeix, Jacques A. Ferland, "Algorithms for generalized fractional programming," *Mathematical Programming,* vol. 52, pp. 191-207, 1991.

[17] H. Benson, "Global optimization of non-linear sums of ratios," *Journal of Mathematical Analysis and Applications,* vol. 263, pp. 301-315, 2001.

[18] H. Benson, "Using concave envelopes to globally solve the non-linear sum of ratios problem," *Journal of Global Optimization,* vol. 22, pp. 343-364, 2002.

[19] H. Benson, "Global optimization algorithm for the non-linear sum of ratios problem," *Journal of Optimization Theory and Applications,* vol. 112, pp. 1-29, 2002.

[20] Y. Dai, J. Shi, Sh. Wang, "A conical partition algorithm for maximizing the sum of several dc ratios," *Journal of Global Optimization,* vol. 31, pp. 253-270, 2005.

[21] Y. Dai, J. Shi, and Y. Yamamoto, "Global optimization problem with several reverse convex constraints and its application to the out-of-roundness problem," *of the Operations Research Society of Japan,* vol. 39, pp. 356-371, 1996.

[22] Mirjam. Dur, Reiner. Horst & Nguyen. Van Thoai, "Solving sum of-ratios fractional programs using efficient points," *Journal of Mathematical Programming and Operations Research,* vol. 49, no. 5-6, pp. 447-466, 2001.

[23] Yan-Jun Wang, Ke-Cun Zhang, "Global optimization of the non-linear sum of ratios problem," *Applied Mathematics and Computation*, vol. 158, pp. 319-330, 2004.

[24] Danny Z. Chen, Ovidiu Daescu, Yang Dai, Naoki Katoh, Xiaodong Wu & Jinhui Xu, "Efficient Algorithms and Implementations for Optimizing the Sum of Linear Fractional Functions with Applications," *Journal of Combinatorial Optimization*, vol. 9, pp. 69-90, 2005.

[25] Pei-Ping Shen, Chun-Feng Wang, "Global optimization for sum of linear ratios problem with coefficients," *Applied Mathematics and Computation*, vol. 176, pp. 219-229, 2006.

[26] Ying Ji, Ke-Cun Zhang, Shao-Jian Qu, "A deterministic global optimization algorithm," *Applied Mathematics and Computation*, vol. 185, pp. 382-387, 2007.

[27] Jean-Pierre Crouzeix, Jacques A. Ferland & Hien Van Nguyen, "Revisiting Dinkelbach-type algorithms for generalized fractional programs," *OPSEARCH*, vol. 45, no. 2, pp. 97-110, 2008.

[28] Wang Chun-Feng, Shen Pei-Ping, "A global optimization algorithm for linear fractional programming," *Applied Mathematics and Computation*, vol. 204, pp. 281-287, 2008.

[29] Zhongmin Fang, Zhiya Chen, "A global optimization method for the non-linear sum of ratios problem," in *Third International Joint Conference on Computational Science and Optimization*, 2010.

[30] Y. Jong, "Practical Global Optimization Algorithm for the sum- of-ratios problem," 22 Jul 2012. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1207/1207.1153.pdf. [Accessed 20 Jul 2020].

[31] Y. Jong, "An Efficient Global Optimization Algorithm for Nonlinear Sum-of-Ratios Problem," 3 May 2012. [Online]. Available: http://www.optimization-online.org/DB_FILE/2012/08/3586.pdf. [Accessed 20 July 2020].

[32] Yuelin Gao and Siqiao Jin, "A Global Optimization Algorithm for Sum of Linear Ratios Problem," *Journal of Applied Mathematics*, 2013.

[33] John Gunnar Carlsson, Jianming Shi, "A linear relaxation algorithm for solving the sum-of-linear-ratios problem with lower dimension," *Operations Research Letters*, vol. 41, no. 4, pp. 381-389, 2013.

[34] HongweiJia, Zhankui Wang, Yongqiang Chen, "Global optimization algorithm for the sum of generalized polynomial ratios problem," *Applied Mathematical Modelling,* vol. 37, no. 1-2, pp. 187-197, 2013.

[35] Li Jin and Xue-Ping Hou, "Global Optimization for a Class of Nonlinear Sum of Ratios Problem," *Mathematical Problems in Engineering,* 2014.

[36] Hongwei Jiao & Sanyang Liu, "An Efficient Algorithm for Quadratic Sum-of-Ratios Fractional Programs Problem," *Numerical Functional Analysis and Optimization,* vol. 38, no. 11, pp. 1426-1445, 2017.

[37] Tatiana V. Gruzdeva, Alexander S. Strekalovsky, "On solving the sum-of-ratios problem," *Applied Mathematics and Computation,* vol. 318, pp. 260-269, 2018.

[38] K. Shen and W. Yu, "Fractional Programming for Communication Part I," *IEEE Transactions on Signal Processing,* vol. 66, no. 10, pp. 2616-2630, 2018.

[39] K. Shen and W. Yu, "Fractional Programming for Communication Systems—Part II: Uplink Scheduling via Matching," *IEEE Transactions on Signal Processing,* vol. 66, no. 10, pp. 2631-2644, 2018.

[40] C. R., "Differential and Integral Calculus," *New York: Interscience,* vol. 2, 1962.

[41] Z.-Q. Luo, W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE Journal on Selected Areas in Communications,* vol. 24, no. 8, pp. 1426-1438, 2006.

[42] M. Chiang, "Geometric programming for communication systems," *Foundations and Trends in Communications and Information Theory,* vol. 2, no. 1, pp. 1-154, 2005.

[43] A. Fehske, J. Malmodin, G. Biczók, G. Fettweis, "The global footprint of mobile communications: The ecological and economic perspective," *IEEE Communications Magazine,* vol. 49, no. 8, pp. 55-62, 2011.

[44] Ericsson, "More Than 50 Billion Connected Devices," Ericsson White Paper, 2011.

[45] A.Zappone, E.A.Jorswieck, "Energy-efficient resource allocation in future wireless networks by sequential fractional programming," *Digital Signal Processing,* vol. 60, pp. 324-337, 2017.

[46] A. Zappone, E. Jorswieck, "Energy efficiency in wireless networks via fractional programming theory," *Found. Trends Commun. Inf. Theory,* vol. 11, no. 3-4, 2015.

[47] Yongyu Dai, Xiaodai Dong, "Power allocation for multi-pair massive MIMO two-way A.F. relaying with linear processing," *IEEE Transaction on Wireless Communication*, vol. 15, no. 9, pp. 5932-5946, 2016.

[48] Cunha Pan, Wei Xu, Wence Zhang, Jiangzhou Wang, Hong Ren, and Ming Chen, "Weighted sum energy efficiency maximization in ad hoc networks," *IEEE Wireless Communication Letters,* vol. 4, no. 3, pp. 233-236, 2015.

[49] E. Sharma, S. S. Chauhan and R. Budhiraja, "Weighted Sum Energy Efficiency Optimization for Massive MIMO Two-Way Half-Duplex A.F. Relaying," *IEEE Wireless Communications Letters,* vol. 8, no. 1, pp. 25-28, 2019.

[50] Li Ping Qian, Ying Jun (Angela) Zhang, Jianwei Huang, "MAPEL: Achieving Global Optimality for a Non-Convex Wireless Power Control Problem," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1553-1563, 2009.

[51] H. Weingarten, Y. Steinberg, and S. Shamai, "The Capacity Region of the Gaussian Multiple-Input Multiple-Output Broadcast Channel," *IEEE TRANSACTIONS ON INFORMATION THEORY,* vol. 52, no. 9, pp. 3936-3964, 2006.

[52] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and Duality," *IEEE Journal of Special Topics in Signal Processing,* vol. 2, no. 1, pp. 57-73, 2008.

[53] Satya Krishna Joshi, Pradeep Chathuranga Weeraddana, Marian Codreanu, and Matti Latva-aho, "Weighted Sum-Rate Maximization for MISO Downlink Cellular Networks via Branch and Bound," *IEEE Transactions on Signal Processing,* vol. 60, no. 4, pp. 2090-2095, 2012.

[54] L. Tran, M. F. Hanif, A. Tolli, and M. Juntti, "Fast Converging Algorithm for Weighted Sum Rate Maximization in Multicell MISO Downlink," *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 872-875, 2012.