

A new algorithm for generalized fractional programs ¹

A.I. Barros ^a, J.B.G. Frenk ^{b,*}, S. Schaible ^c, S. Zhang ^b

^a D.E.I.O., Faculdade de Ciências, Universidade de Lisboa, Campo Grande, 1700 Lisboa, Portugal

^b Econometric Institute, Erasmus University, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands

^c University of California at Riverside, Riverside, CA 92521, USA

Received 2 February 1994

Abstract

A new dual problem for convex generalized fractional programs with no duality gap is presented and it is shown how this dual problem can be efficiently solved using a parametric approach. The resulting algorithm can be seen as “dual” to the Dinkelbach-type algorithm for generalized fractional programs since it approximates the optimal objective value of the dual (primal) problem from below. Convergence results for this algorithm are derived and an easy condition to achieve superlinear convergence is also established. Moreover, under some additional assumptions the algorithm also recovers at the same time an optimal solution of the primal problem. We also consider a variant of this new algorithm, based on scaling the “dual” parametric function. The numerical results, in case of quadratic-linear ratios and linear constraints, show that the performance of the new algorithm and its scaled version is superior to that of the Dinkelbach-type algorithms. From the computational results it also appears that contrary to the primal approach, the “dual” approach is less influenced by scaling.

Keywords: Fractional programming; Generalized fractional programming; Dinkelbach-type algorithms; Quasi-convexity; Karush–Kuhn–Tucker conditions; Duality

1. Introduction

Fractional programming, i.e., the minimization of a ratio of two functions subject to constraints, has been studied extensively during the last several decades [1,8,27,28]. In

* Corresponding author.

¹ This research was carried out at the Econometric Institute, Erasmus University, Rotterdam, the Netherlands and was supported by J.N.I.C.T. (Portugal) under contract BD/707/90-RM.

the 1980s the focus has shifted towards multi-ratio optimization problems. One of the major types is the problem of minimizing the largest of several ratios of functions. These so-called generalized fractional programs arise in economic equilibrium problems, in management applications of goal programming and multiobjective programming involving ratios of functions, and in rational approximation in numerical analysis [10].

Algorithmic and computational results for single-ratio fractional programs can be found in [17,18,28] and in the literature cited therein. Various algorithms in generalized fractional programming are surveyed in [9] whereas computational experience with some of these algorithms is reported in [5,7,14]. Meanwhile also a duality theory for generalized fractional programs has been developed [1,8].

The purpose of this paper is to introduce a new dual problem for convex generalized fractional programs and an algorithm to solve this problem. The main feature of this algorithm is that at each iteration a single-ratio fractional programming problem is solved and the optimal objective value of this fractional programming problem provides a lower bound on the optimal objective value of the original generalized fractional program. Following the strategy used to derive the Dinkelbach-type-2 algorithm we will also propose a variant of this “dual” algorithm.

The paper is organized in the following way. We start by presenting a short overview on the basic algorithms for generalized fractional programs. In Section 3 the new algorithm is introduced and convergence results are discussed. A variant of this algorithm, based on scaling of the “dual” parametric function, is studied in Section 4. Finally some computational results are presented comparing the performance of the new algorithms with the Dinkelbach-type approaches [11,12].

2. Algorithms for generalized fractional programming

Let $\mathcal{X} \subset \mathbb{R}^n$ be compact and assume that the functions $f_i, g_i: \mathcal{S} \rightarrow \mathbb{R}$, $i \in I := \{1, \dots, m\}$, $m \geq 1$, are continuous where \mathcal{S} is an open set containing \mathcal{X} . Also let $g_i(\mathbf{x}) > 0$ for every $\mathbf{x} \in \mathcal{X}$ and $i \in I$. We consider the generalized fractional program

$$(P) \quad \inf_{\mathbf{x} \in \mathcal{X}} \max_{i \in I} \frac{f_i(\mathbf{x})}{g_i(\mathbf{x})}.$$

Since the function $\mathbf{x} \mapsto \max_{i \in I} f_i(\mathbf{x})/g_i(\mathbf{x})$ is finite-valued and continuous on \mathcal{X} and $\mathcal{X} \subseteq \mathbb{R}^n$ is compact, the optimization problem (P) has an optimal solution. Clearly, for $m = 1$ problem (P) reduces to a single-ratio fractional programming problem [13]. To solve (P), we consider the following parametric problem:

$$(P_\mu) \quad F(\mu) = \inf_{\mathbf{x} \in \mathcal{X}} \left\{ \max_{i \in I} \{f_i(\mathbf{x}) - \mu g_i(\mathbf{x})\} \right\}.$$

By a similar argument as for (P), problem (P_μ) also has an optimal solution, and both optimization problems are related to each other by the following result [11].

Lemma 2.1. *If \mathcal{X} is compact, $m \geq 1$ and $-\infty < \mu < +\infty$, then*

- (a) $F(\mu) < +\infty$, F is decreasing and continuous;
- (b) (P) and (P_μ) have optimal solutions;
- (c) the optimal objective value μ^* of (P) is finite and $F(\mu^*) = 0$;
- (d) $F(\mu) = 0$ implies $\mu = \mu^*$;
- (e) if $F(\mu) = 0$ then (P) and (P_μ) have the same set of optimal solutions.

By Lemma 2.1 it is clear that solving (P) can be achieved by finding a solution of the equation $F(\mu) = 0$. Based on this observation, the Dinkelbach-type algorithm proposed in [11] solves at each step a subproblem (P_μ) , and by doing so it creates a nonincreasing sequence μ_k , $k \geq 1$, converging from above to the optimal objective value μ^* of problem (P). More precisely, at the k th step of this procedure μ_{k+1} is taken as the root of the equation $G_k(\mu) = 0$ with $G_k: \mathbb{R} \rightarrow \mathbb{R}$ a linear (piecewise linear) upper envelope of the function F for $m = 1$ ($m > 1$) and $G_k(\mu_k) = F(\mu_k) \leq 0$. The Dinkelbach-type algorithm can now be summarized as follows.

Algorithm 2.1 (Dinkelbach-type algorithm).

- Step 0. Take $x_0 \in \mathcal{X}$, compute $\mu_1 := \max_{i \in I} f_i(x_0)/g_i(x_0)$ and let $k := 1$;
- Step 1. Determine $x_k := \operatorname{argmin}_{x \in \mathcal{X}} \{\max_{i \in I} \{f_i(x) - \mu_k g_i(x)\}\}$;
- Step 2. **If** $F(\mu_k) = 0$
 - Then** x_k is an optimal solution of (P) with value μ_k and **Stop**.
 - Else GoTo** Step 3;
- Step 3. Let $\mu_{k+1} := \max_{i \in I} f_i(x_k)/g_i(x_k)$;
 Let $k := k + 1$, and **GoTo** Step 1.

Obviously, it is only useful to apply the above procedure if every subproblem (P_μ) is easier to solve than the original problem (P). Furthermore it is also clear that a single-ratio fractional program, i.e., $m = 1$, is easier to solve than a generalized fractional program, i.e., $m > 1$. In [3] it is shown that the above basic algorithm is a special case of a cutting plane algorithm on the space \mathcal{X} .

A refinement of the Dinkelbach-type algorithm was later proposed by Crouzeix et al. [12] and independently by Flachs [15]. The main idea behind this variant consists in trying to make the parametric function concave in a neighborhood of the optimal value. In order to achieve this, Crouzeix et al. [12] propose the following reformulation of (P):

$$\inf_{x \in \mathcal{X}} \max_{i \in I} \frac{f_i(x)/g_i(x_*)}{g_i(x)/g_i(x_*)},$$

where x_* denotes an optimal solution of (P). The associated parametric problem is given by

$$\inf_{x \in \mathcal{X}} \left\{ \max_{i \in I} \left\{ \frac{f_i(x) - \mu g_i(x)}{g_i(x_*)} \right\} \right\}.$$

In practice, since an optimal solution of (P) is not known a priori, the current iteration point is used as an approximation of x_* . Hence, in the Dinkelbach-type-2 algorithm the following parametric problem is considered:

$$(P_{\mu}^{(k-1)}) \quad F^{(k-1)}(\mu) = \inf_{x \in \mathcal{X}} \left\{ \max_{i \in I} \left\{ \frac{f_i(x) - \mu g_i(x)}{g_i(x_{k-1})} \right\} \right\},$$

with $x_{k-1} \in \mathcal{X}$ the last iteration point. By our conditions this problem has an optimal solution and so, the Dinkelbach-type-2 algorithm can be described as follows.

Algorithm 2.2 (Dinkelbach-type-2 algorithm).

Step 0'. Take $x_0 \in \mathcal{X}$, compute $\mu_1 := \max_{i \in I} f_i(x_0)/g_i(x_0)$ and let $k := 1$;

Step 1'. Determine:

$$x_k := \operatorname{argmin}_{x \in \mathcal{X}} \left\{ \max_{i \in I} \left\{ \frac{f_i(x) - \mu_k g_i(x)}{g_i(x_{k-1})} \right\} \right\}$$

Step 2'. If $F^{(k-1)}(\mu_k) = 0$

Then x_k is an optimal solution of (P) with value μ_k and **Stop**.

Else GoTo Step 3';

Step 3'. Let $\mu_{k+1} := \max_{i \in I} f_i(x_k)/g_i(x_k)$;

Let $k := k + 1$, and **GoTo** Step 1'.

Based on the Dinkelbach-type approaches and their geometrical interpretation, several interval type algorithms have been proposed. A thorough overview of these algorithms can be found in the survey by Crouzeix and Ferland [9].

3. The dual problem and how to solve it

In this section we propose a new dual problem for (P) with no duality gap and introduce at the same time an algorithm to solve simultaneously this dual and the corresponding primal problem. This algorithm is ‘‘dual’’ to the Dinkelbach-type approach since it creates a nondecreasing sequence μ_k , $k \geq 1$, converging from below to μ^* . We assume that the functions $f_i: \mathcal{S} \rightarrow \mathbb{R}$, $i \in I$, are convex on the compact convex set \mathcal{X} and $g_i: \mathcal{S} \rightarrow \mathbb{R}$, $i \in I$, are positive and concave on \mathcal{X} . In addition, either the functions f_i , $i \in I$, are nonnegative on \mathcal{X} or the functions g_i , $i \in I$, are affine on \mathcal{X} . Notice that the Dinkelbach-type algorithm proposed in [11] leads to convex programs as subproblems under the same assumptions. Observe also that these assumptions include the important case of generalized linear fractional programs with a bounded feasible region.

If $f(x)^T := (f_1(x), \dots, f_m(x))$ and $g(x)^T := (g_1(x), \dots, g_m(x))$, then it follows by the quasiconvexity of the function $q: \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}$ given by $q(z) := z_1/z_2$ that

$$\max_{i \in I} \frac{f_i(x)}{g_i(x)} = \max_{y \in \Sigma} \frac{y^T f(x)}{y^T g(x)} \tag{1}$$

for every $\mathbf{x} \in \mathcal{X}$ and $\Sigma := \{\mathbf{y} \in \mathbb{R}^m: \mathbf{y} \geq \mathbf{0}, \sum_{i \in I} y_i = 1\}$. This is a direct consequence of the property that a quasiconvex function attains its maximum in a vertex of a convex polyhedron [1]. Moreover, by the assumptions on the vector functions \mathbf{f} and \mathbf{g} we obtain that the function $\mathbf{x} \mapsto \mathbf{y}^T \mathbf{f}(\mathbf{x}) / \mathbf{y}^T \mathbf{g}(\mathbf{x})$ is quasiconvex on \mathcal{X} for every $\mathbf{y} \in \Sigma$, while the function $\mathbf{y} \mapsto \mathbf{y}^T \mathbf{f}(\mathbf{x}) / \mathbf{y}^T \mathbf{g}(\mathbf{x})$ is quasiconcave on Σ for every $\mathbf{x} \in \mathcal{X}$. Hence, using the compactness of the convex sets \mathcal{X} and Σ it follows from Sion’s minimax theorem [30] that

$$\min_{\mathbf{x} \in \mathcal{X}} \left\{ \max_{\mathbf{y} \in \Sigma} \frac{\mathbf{y}^T \mathbf{f}(\mathbf{x})}{\mathbf{y}^T \mathbf{g}(\mathbf{x})} \right\} = \max_{\mathbf{y} \in \Sigma} \left\{ \min_{\mathbf{x} \in \mathcal{X}} \frac{\mathbf{y}^T \mathbf{f}(\mathbf{x})}{\mathbf{y}^T \mathbf{g}(\mathbf{x})} \right\}, \tag{2}$$

and so by (1) and (2) we obtain

$$\min_{\mathbf{x} \in \mathcal{X}} \left\{ \max_{i \in I} \frac{f_i(\mathbf{x})}{g_i(\mathbf{x})} \right\} = \max_{\mathbf{y} \in \Sigma} \left\{ \min_{\mathbf{x} \in \mathcal{X}} \frac{\mathbf{y}^T \mathbf{f}(\mathbf{x})}{\mathbf{y}^T \mathbf{g}(\mathbf{x})} \right\}. \tag{3}$$

Let $c: \Sigma \rightarrow \mathbb{R}$ be defined by

$$c(\mathbf{y}) := \min_{\mathbf{x} \in \mathcal{X}} \frac{\mathbf{y}^T \mathbf{f}(\mathbf{x})}{\mathbf{y}^T \mathbf{g}(\mathbf{x})}.$$

By the compactness of \mathcal{X} and the continuity of the function $h: \mathcal{X} \times \Sigma \rightarrow \mathbb{R}$ given by $h(\mathbf{x}, \mathbf{y}) = \mathbf{y}^T \mathbf{f}(\mathbf{x}) / (\mathbf{y}^T \mathbf{g}(\mathbf{x}))$, this implies that c is continuous on Σ [22]. Moreover, the function c is semistrictly quasiconcave [1], since it is the infimum of semistrictly quasiconcave functions $h(\cdot, \mathbf{y})$ indexed by \mathbf{x} . Thus, by (3), we need to solve the quasiconcave optimization problem:

$$(Q) \quad \max_{\mathbf{y} \in \Sigma} c(\mathbf{y}),$$

where a local maximum is a global maximum [1]. Notice that (Q) corresponds to a new ‘‘dual’’ problem of (P). In fact, while in the standard dual problem of a convex generalized fractional programming problem [1,8,10,19,31] a part of the constraint set is ‘‘dualized’’, problem (Q) can be seen as a ‘‘partial dual’’ program of the generalized fractional program (P), since it only ‘‘dualizes’’ the ratios.

Observe, by (3), that there exists some $\mathbf{y}_* \in \Sigma$ with $c(\mathbf{y}_*) = \mu^*$. This does not mean that any optimal solution $\mathbf{x}^* \in \mathcal{X}$ associated with the single-ratio fractional programming problem $\min_{\mathbf{x} \in \mathcal{X}} \mathbf{y}_*^T \mathbf{f}(\mathbf{x}) / \mathbf{y}_*^T \mathbf{g}(\mathbf{x})$ is also an optimal solution of (P). However, for any optimal solution $\mathbf{x}^* \in \mathcal{X}$ of (P) and $\mathbf{y}_* \in \Sigma$ of (Q) it is easy to see, using

$$\frac{\mathbf{y}_*^T \mathbf{f}(\mathbf{x}_*)}{\mathbf{y}_*^T \mathbf{g}(\mathbf{x}_*)} \leq \max_{i \in I} \frac{f_i(\mathbf{x}_*)}{g_i(\mathbf{x}_*)} = c(\mathbf{y}_*),$$

that \mathbf{x}_* is an optimal solution of the optimization problem associated with $c(\mathbf{y}_*)$. Moreover, it follows immediately that $c(\mathbf{y}) \leq \mu^*$ for every $\mathbf{y} \in \Sigma$, and so an iterative procedure solving (Q) approximates μ^* from below. Although not necessary, as known from subgradient-type algorithms for convex programming [29], it would be a nice

feature if an algorithm solving (Q) would satisfy the descent property. Hence, with y_k the present nonoptimal iteration point, the next point y_{k+1} should belong to the strict upper level set

$$\mathcal{U}_c^\circ(c(y_k)) := \{y \in \Sigma : c(y) > c(y_k)\}.$$

Introducing also

$$\mathcal{U}_c(c(y_k)) := \{y \in \Sigma : c(y) \geq c(y_k)\},$$

the following result motivates the choice of the next iteration point.

Lemma 3.1. *Let $F : \Sigma \times \mathbb{R} \rightarrow \mathbb{R}$ be given by*

$$F(y, \mu) := \min_{x \in \mathcal{X}} \{y^T(f(x) - \mu g(x))\}. \tag{4}$$

For $\hat{y} \in \Sigma$ we have $\mathcal{U}_c^\circ(c(\hat{y})) = \{y \in \Sigma : F(y, c(\hat{y})) > 0\}$ and $\mathcal{U}_c(c(\hat{y})) = \{y \in \Sigma : F(y, c(\hat{y})) \geq 0\}$.

Proof. We first consider the case that $\mathcal{U}_c^\circ(c(\hat{y}))$ is nonempty, i.e., $\hat{y} \in \Sigma$ is nonoptimal for (Q). Let $y \in \mathcal{U}_c^\circ(c(\hat{y}))$. Then $c(y) > c(\hat{y})$ and from Lemma 2.1 and $g(x) > 0$ for every $x \in \mathcal{X}$ we see

$$\begin{aligned} F(y, c(\hat{y})) &= \min_{x \in \mathcal{X}} \{y^T(f(x) - c(\hat{y})g(x))\} \\ &> \min_{x \in \mathcal{X}} \{y^T(f(x) - c(y)g(x))\} = 0. \end{aligned} \tag{5}$$

Conversely, if $F(y, c(\hat{y})) > 0$ and $y \in \Sigma$, then using again Lemma 2.1 it follows that $c(y) > c(\hat{y})$ which concludes the proof for the nonempty case.

On the other hand, if $\mathcal{U}_c^\circ(c(\hat{y}))$ is empty, then we know that $c(y) \leq c(\hat{y})$ for every $y \in \Sigma$, and hence by Lemma 2.1 the set $\{y \in \Sigma : F(y, c(\hat{y})) > 0\}$ is also empty.

Finally the last equality can be proved in a similar way as the first part of this proof, and so we omit it. \square

Let y_k be the present iteration point. The above result and Lemma 2.1 suggest that the ‘‘best’’ possible choice for the next iteration point y_{k+1} is given by

$$F(y_{k+1}, c(y_k)) := \max_{y \in \Sigma} F(y, c(y_k)). \tag{6}$$

Observe that the above optimization problem can be seen as the parametric problem of (Q) with parameter $c(y_k)$. After having solved (6) we compute $c(y_{k+1})$ and continue with k replaced by $k + 1$. This yields the following algorithm.

Algorithm 3.1 (‘‘Dual’’ algorithm).

Step 0. Take $y_0 \in \Sigma$, compute $c(y_0) = \min_{x \in \mathcal{X}} y_0^T f(x) / y_0^T g(x)$ and let $k := 1$;

Step 1. Determine $y_k := \operatorname{argmax}_{y \in \Sigma} F(y, c(y_{k-1}))$;

Step 2. If $F(y_k, c(y_{k-1})) = 0$

Then y_{k-1} is an optimal solution with value $c(y_{k-1})$ and **Stop**.

Else **GoTo** Step 3;

Step 3. Compute $c(y_k)$;

Let $k := k + 1$, and **GoTo** Step 1.

Notice that, a primal optimal solution of (P) can be found by solving the parametric problem (P_μ) with $\mu = c(y_{k-1})$.

By our assumptions on f and g we may apply Von Neumann’s min–max theorem [24], and so

$$\begin{aligned}
 F(y_{k+1}, c(y_k)) &= \max_{y \in \Sigma} F(y, c(y_k)) \\
 &= \max_{y \in \Sigma} \left\{ \min_{x \in \mathcal{X}} \{y^T(f(x) - c(y_k)g(x))\} \right\} \\
 &= \min_{x \in \mathcal{X}} \left\{ \max_{y \in \Sigma} \{y^T(f(x) - c(y_k)g(x))\} \right\} \\
 &= \min_{x \in \mathcal{X}} \left\{ \max_{i \in I} \{f_i(x) - c(y_k)g_i(x)\} \right\} \\
 &= F(c(y_k)).
 \end{aligned} \tag{7}$$

Observe that, even in the case that the conditions of Von Neumann’s min–max theorem do not hold, $F(c(y_k))$ is always an upper bound for $F(y_{k+1}, c(y_k))$.

For a geometrical interpretation of this ‘‘lower bounding’’ algorithm we introduce for each fixed $y \in \Sigma$ the function $F_y : \mathbb{R} \rightarrow \mathbb{R}$ given by

$$F_y(\mu) := \min_{x \in \mathcal{X}} \{y^T(f(x) - \mu g(x))\}. \tag{8}$$

Clearly, for every $\mu \in \mathbb{R}$ we have that $F_y(\mu) \leq F(\mu)$ with $y \in \Sigma$, while for $\mu = c(y_k)$ it was shown using Von-Neumann’s min-max theorem that $F_{y_{k+1}}(\mu) = F(\mu)$. Moreover, to determine $c(y_{k+1})$ is equivalent to finding the root of $F_{y_{k+1}}(\mu) = 0$, and so this yields the geometrical interpretation given in Fig. 1.

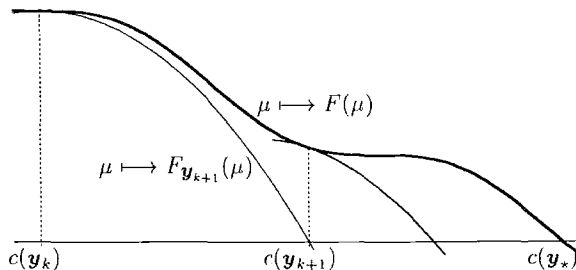


Fig. 1. New algorithm.

To prove the convergence of this algorithm we need to investigate the behavior of the function $F_y : \mathbb{R} \rightarrow \mathbb{R}$. Since by (8) this function is the minimum of a set of affine functions, it is concave, and so by [25 Corollary 10.1.1.] it is continuous on \mathbb{R} . Also, by [25, Theorem 23.4] the subgradient set $\partial(-F_y)(\mu)$ of the convex function $-F_y : \mathbb{R} \rightarrow \mathbb{R}$ at the point μ is nonempty. Remember that $d \in \mathbb{R}$ is a subgradient of the function $-F_y$ at the point μ if and only if

$$F_y(\mu + t) \leq F_y(\mu) - td, \tag{9}$$

for every $t \in \mathbb{R}$.

The next result characterizes the subgradient set $\partial(-F_y)(\mu)$. Although this result is a special case of a more general result given by [26, Theorem 7.2] or [16, Theorem VI.4.4.2] we give an elementary proof for completeness. This is possible due to the fact that F_y is defined on \mathbb{R} . However, before mentioning this result we introduce for fixed $y \in \Sigma$ the set $\mathcal{X}_y(\mu)$ of optimal solutions of the optimization problem $\min_{x \in \mathcal{X}} \{y^T(f(x) - \mu g(x))\}$, i.e.,

$$\mathcal{X}_y(\mu) := \{x \in \mathcal{X} : y^T(f(x) - \mu g(x)) = F_y(\mu)\}. \tag{10}$$

Clearly, this set is nonempty. Also, by the continuity of the vector-valued functions f and g it must be closed, and thus by the compactness of \mathcal{X} and $\mathcal{X}_y(\mu) \subseteq \mathcal{X}$ it is compact. Finally, if $\mu \geq 0$ or $\mu \in \mathbb{R}$ whenever g_i are positive affine functions then the function $x \rightarrow y^T(f(x) - \mu g(x))$ is convex on \mathcal{X} due to f convex and g concave (affine), and this implies that $\mathcal{X}_y(\mu) \subseteq \mathcal{X}$ is also convex for appropriate values of μ .

Lemma 3.2. *For every fixed $y \in \Sigma$ and $\mu \in \mathbb{R}$ it follows that*

$$\partial(-F_y)(\mu) = \left[\inf_{x \in \mathcal{X}_y(\mu)} \{y^T g(x)\}, \sup_{x \in \mathcal{X}_y(\mu)} \{y^T g(x)\} \right].$$

Proof. Since every convex combination of subgradients is again a subgradient, we only need to verify for the inclusion \supseteq that the left and the right endpoint of the above compact interval are subgradients.

For $d = \sup_{x \in \mathcal{X}_y(\mu)} \{y^T g(x)\}$ we obtain by the compactness of $\mathcal{X}_y(\mu)$ and the continuity on \mathcal{X} of the function $x \rightarrow y^T g(x)$ that there exists some $\bar{x} \in \mathcal{X}_y(\mu)$ satisfying $d = y^T g(\bar{x})$. Hence,

$$\begin{aligned} -td &= y^T(f(\bar{x}) - (\mu + t)g(\bar{x})) - y^T(f(\bar{x}) - \mu g(\bar{x})) \\ &\geq F_y(\mu + t) - F_y(\mu) \end{aligned} \tag{11}$$

and so by (9) it follows that d is a subgradient. A similar proof holds for the left endpoint and so the inclusion \supseteq is verified.

To verify the reverse inclusion \subseteq we first observe the following. Consider the sequence $x_n, n \geq 1$, with $x_n \in \mathcal{X}_y(\mu + 1/n)$. By the compactness of \mathcal{X} the sequence $\{x_n, n \geq 1\}$, has an accumulation point $x_x \in \mathcal{X}$ and so there exists some subsequence

$\{x_{n_l}; l \geq 1\}$ with $\lim_{l \uparrow \infty} x_{n_l} = x_z$. Hence by the continuity of the functions $\mu \mapsto F_y(\mu)$, $x \mapsto y^T f(x)$, and $x \mapsto y^T g(x)$ respectively, it follows that

$$\begin{aligned} F_y(\mu) &= \lim_{l \uparrow \infty} F_y\left(\mu + \frac{1}{n_l}\right) \\ &= \lim_{l \uparrow \infty} \left(y^T f(x_{n_l}) - \left(\mu + \frac{1}{n_l}\right) y^T g(x_{n_l}) \right) \\ &= y^T f(x_z) - \mu y^T g(x_z) \end{aligned} \tag{12}$$

and so $x_z \in \mathcal{X}_y(\mu)$.²

Again, by the continuity of the function $x \mapsto y^T g(x)$ there exists for every $\delta > 0$ some $l_\delta \geq 1$ such that for every $l \geq l_\delta$ the inequality

$$y^T g(x_{n_l}) \leq y^T g(x_z) + \delta \leq \sup_{x \in \mathcal{X}_y(\mu)} \{y^T g(x)\} + \delta$$

holds. From this result it is clear that for any $d \in \partial(-F_y)(\mu)$ and $l \geq l_\delta$ we obtain

$$\begin{aligned} -\frac{1}{n_l} d &\geq F_y\left(\mu + \frac{1}{n_l}\right) - F_y(\mu) \geq -\frac{1}{n_l} y^T g(x_{n_l}) \\ &\geq -\frac{1}{n_l} \left(\sup_{x \in \mathcal{X}_y(\mu)} \{y^T g(x)\} + \delta \right). \end{aligned} \tag{13}$$

Multiplying this inequality by $-n_l$ yields $d \leq \sup_{x \in \mathcal{X}_y(\mu)} \{y^T g(x)\} + \delta$. Since $\delta > 0$ can be arbitrarily chosen, we obtain that

$$d \leq \sup_{x \in \mathcal{X}_y(\mu)} \{y^T g(x)\}.$$

By considering a sequence $x_n \in \mathcal{X}_y(\mu - 1/n)$ and applying a similar proof one can show that $d \geq \inf_{x \in \mathcal{X}_y(\mu)} \{y^T g(x)\}$, thus completing the inclusion \subseteq . \square

As already observed, we may replace inf, respectively sup, in the interval mentioned in Lemma 3.2 by min, respectively max. Moreover, since $g(x) > 0$ for every $x \in \mathcal{X}$, we obtain by the subgradient inequality given by (9) and Lemma 3.2 that the function $F_y: \mathbb{R} \rightarrow \mathbb{R}$ is decreasing³. Denote now by k^* the number of times that the main step was started by the algorithm. Clearly if $k^* = +\infty$ the algorithm does not stop while for k^* finite it follows that $F(y_{k^*}, c(y_{k^* - 1})) = 0$. Before mentioning the next result we introduce

$$\Delta_k(y) := \max\{y^T g(x) : x \in \mathcal{X}_y(c(y_k))\}$$

² By [21 Proposition 2.1.4] this means that for fixed $y \in \Sigma$ the point-to-set mapping $\mu \mapsto \mathcal{X}_y(\mu)$ is upper semicontinuous.

³ Observe that the same result also follows from (8).

and

$$\begin{aligned} \delta_{k+1} &:= \min \left\{ \mathbf{y}_{k+1}^T \mathbf{g}(\mathbf{x}) : \mathbf{x} \text{ solves } \min_{\mathbf{x} \in \mathcal{X}} \frac{\mathbf{y}_{k+1}^T \mathbf{f}(\mathbf{x})}{\mathbf{y}_{k+1}^T \mathbf{g}(\mathbf{x})} \right\} \\ &= \min \left\{ \mathbf{y}_{k+1}^T \mathbf{g}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}_{\mathbf{y}_{k+1}}(c(\mathbf{y}_{k+1})) \right\}. \end{aligned} \tag{14}$$

It follows by Lemma 3.2 that

$$\Delta_k(\mathbf{y}) \in \partial(-F_{\mathbf{y}})(c(\mathbf{y}_k)) \quad \text{and} \quad \delta_{k+1} \in \partial(-F_{\mathbf{y}_{k+1}})(c(\mathbf{y}_{k+1})).$$

Theorem 3.1. *The sequence \mathbf{y}_k , $0 \leq k < k^*$, does not contain optimal solutions of (Q) and the corresponding function values $c(\mathbf{y}_k)$, $0 \leq k < k^*$, are increasing. Moreover, if k^* is finite, then $c(\mathbf{y}_{k^*}) = \mu^*$ while for $k^* = +\infty$ every accumulation point of the sequence \mathbf{y}_k , $k \geq 0$, is an optimal solution of (Q). Finally, if $k^* = +\infty$ and \mathbf{y}_* is an optimal solution of (Q), then*

$$0 \leq \mu^* - c(\mathbf{y}_{k+1}) \leq \left(1 - \frac{\Delta_k(\mathbf{y}_*)}{\delta_{k+1}} \right) (\mu^* - c(\mathbf{y}_k)) \tag{15}$$

holds for every $k \geq 0$.

Proof. Using Lemma 3.1, it follows that \mathbf{y}_k is nonoptimal if and only if $F(\mathbf{y}_{k+1}, c(\mathbf{y}_k)) > 0$. Moreover, by the same lemma we obtain that $c(\mathbf{y}_{k+1}) > c(\mathbf{y}_k)$ if \mathbf{y}_k is nonoptimal, and so the first part of the theorem is proved.

Observe for k^* finite that $F(\mathbf{y}_{k^*}, c(\mathbf{y}_{k^*-1})) = 0$, and again by Lemma 3.1 it follows that \mathbf{y}_{k^*} solves (Q). Hence by (3) we have $c(\mathbf{y}_{k^*}) = \mu^*$.

To verify the last part of the result, notice that $c(\mathbf{y}_k)$, $k \geq 0$, is increasing for $k^* = +\infty$, and since $c(\mathbf{y}_k) \leq \mu^* < \infty$ for every $k \geq 0$, it must follow that $\lim_{k \uparrow \infty} c(\mathbf{y}_k)$ exists and is finite-valued. Moreover, by Lemma 3.2 and (9) we obtain for every optimal solution \mathbf{y}_* of (Q) that

$$F_{\mathbf{y}_*}(c(\mathbf{y}_*)) - F_{\mathbf{y}_*}(c(\mathbf{y}_k)) \leq -(c(\mathbf{y}_*) - c(\mathbf{y}_k)) \Delta_k(\mathbf{y}_*).$$

Since $F_{\mathbf{y}_*}(c(\mathbf{y}_*)) = 0$ this implies that

$$\begin{aligned} F_{\mathbf{y}_{k+1}}(c(\mathbf{y}_k)) &= \max_{\mathbf{y} \in \Sigma} F_{\mathbf{y}}(c(\mathbf{y}_k)) \geq F_{\mathbf{y}_*}(c(\mathbf{y}_k)) \\ &\geq (c(\mathbf{y}_*) - c(\mathbf{y}_k)) \Delta_k(\mathbf{y}_*). \end{aligned} \tag{16}$$

On the other hand, applying again Lemma 3.2 and (9) we obtain

$$\begin{aligned} F_{\mathbf{y}_{k+1}}(c(\mathbf{y}_k)) &= F_{\mathbf{y}_{k+1}}(c(\mathbf{y}_k)) - F_{\mathbf{y}_{k+1}}(c(\mathbf{y}_{k+1})) \\ &\leq (c(\mathbf{y}_{k+1}) - c(\mathbf{y}_k)) \delta_{k+1}. \end{aligned} \tag{17}$$

This implies by (16) and (17) that

$$(c(\mathbf{y}_{k+1}) - c(\mathbf{y}_k)) \delta_{k+1} \geq (c(\mathbf{y}_*) - c(\mathbf{y}_k)) \Delta_k(\mathbf{y}_*) = (\mu^* - c(\mathbf{y}_k)) \Delta_k(\mathbf{y}_*). \tag{18}$$

Since $\Delta_k(\mathbf{y}_*)$ and δ_{k+1} belong to the interval $[\delta, \Delta]$ with

$$\delta := \min_{\mathbf{x} \in \mathcal{X}} \min_{i \in I} g_i(\mathbf{x}) > 0 \quad \text{and} \quad \Delta := \max_{\mathbf{x} \in \mathcal{X}} \max_{i \in I} g_i(\mathbf{x}) < +\infty,$$

it follows by (18) and the existence of $\lim_{k \uparrow \infty} c(\mathbf{y}_k)$ that $\lim_{k \uparrow \infty} c(\mathbf{y}_k) = c(\mathbf{y}_*)$, and so every accumulation point of the sequence $\mathbf{y}_k, k \geq 0$, solves (Q).

Finally, from (18) we obtain

$$\begin{aligned} \mu^* - c(\mathbf{y}_{k+1}) &= \mu^* - c(\mathbf{y}_k) + c(\mathbf{y}_k) - c(\mathbf{y}_{k+1}) \\ &\leq \left(1 - \frac{\Delta_k(\mathbf{y}_*)}{\delta_{k+1}}\right) (\mu^* - c(\mathbf{y}_k)). \quad \square \end{aligned}$$

Clearly, by inequality (15) this algorithm converges at least linearly. In order to improve this convergence rate result we need to investigate the behavior of $1 - \Delta_k(\mathbf{y}_*)/\delta_{k+1}$ as $k \rightarrow \infty$ for an arbitrary optimal solution \mathbf{y}_* of (Q). Let $\delta_z := \limsup_{k \uparrow \infty} \delta_{k+1}$. By the definition of \limsup there exists a subsequence $\mathcal{K} \subseteq \mathbb{N}$ such that $\delta_z = \lim_{k \in \mathcal{K}, k \uparrow \infty} \delta_{k+1}$. Moreover, if we consider the sequence $\{\mathbf{y}_{k+1} : k \in \mathcal{K}\} \subseteq \Sigma$ we can also find, due to the compactness of Σ , a subsequence $\mathcal{K}_1 \subseteq \mathcal{K}$ satisfying $\lim_{k \in \mathcal{K}_1, k \uparrow \infty} \mathbf{y}_{k+1} = \mathbf{y}_*$ with \mathbf{y}_* being an accumulation point. By Theorem 3.1 this accumulation point \mathbf{y}_* is an optimal solution of (Q). Consider now the sequence $1 - \Delta_k(\mathbf{y}_*)/\delta_{k+1}$ for this point \mathbf{y}_* . It is easy to verify that the point-to-set mapping $\mathbf{y} \mapsto \partial(-F_{\mathbf{y}})(c(\mathbf{y}))$ is upper semicontinuous. Hence we obtain, due to $\delta_{k+1} \in \partial(-F_{\mathbf{y}_{k+1}})(c(\mathbf{y}_{k+1}))$, $\lim_{k \in \mathcal{K}_1, k \uparrow \infty} \delta_{k+1} = \delta_z$ and $\lim_{k \in \mathcal{K}_1, k \uparrow \infty} \mathbf{y}_{k+1} = \mathbf{y}_*$ that

$$\delta_z \in \partial(-F_{\mathbf{y}_*})(c(\mathbf{y}_*)). \tag{19}$$

On the other hand, it is clear by Lemma 3.2 that $\Delta_k(\mathbf{y}_*) \in \partial(-F_{\mathbf{y}_*})(c(\mathbf{y}_k))$. Moreover, since the sequence $c(\mathbf{y}_k)$ converges monotonically from below to $c(\mathbf{y}_*)$, it follows by the convexity of the function $\mu \mapsto -F_{\mathbf{y}_*}(\mu)$ and our previous observation $\Delta_k(\mathbf{y}_*) \in \partial(-F_{\mathbf{y}_*})(c(\mathbf{y}_k))$ that $\Delta_k(\mathbf{y}_*) \leq \Delta_{k+1}(\mathbf{y}_*) \leq \dots \leq a_*$ with $a_* \in \partial(-F_{\mathbf{y}_*})(c(\mathbf{y}_*))$. This implies $\lim_{k \uparrow \infty} \Delta_k(\mathbf{y}_*) =: \Delta_z(\mathbf{y}_*)$ exists and by the upper semicontinuity of the point-to-set mapping $\mu \mapsto \partial(-F_{\mathbf{y}_*})(\mu)$ we obtain that $\Delta_z(\mathbf{y}_*) \in \partial(-F_{\mathbf{y}_*})(c(\mathbf{y}_*))$. Since we already observed that $\Delta_z(\mathbf{y}_*) \leq a_*$ for every $a_* \in \partial(-F_{\mathbf{y}_*})(c(\mathbf{y}_*))$, it must follow by Lemma 3.2 that

$$\Delta_z(\mathbf{y}_*) = \min_{\mathbf{x} \in \mathcal{L}_{\mathbf{y}_*}^+(c(\mathbf{y}_*))} \{\mathbf{y}_*^T \mathbf{g}(\mathbf{x})\}. \tag{20}$$

To conclude our analysis of the behavior of the sequence $1 - \Delta_k(\mathbf{y}_*)/\delta_{k+1}$ as $k \rightarrow \infty$, observe by (19) and (20) that

$$0 \leq \limsup_{k \uparrow \infty} \left(1 - \frac{\Delta_k(\mathbf{y}_*)}{\delta_{k+1}}\right) = 1 - \liminf_{k \uparrow \infty} \frac{\Delta_k(\mathbf{y}_*)}{\delta_{k+1}} = 1 - \frac{\Delta_z(\mathbf{y}_*)}{\delta_z} < 1.$$

These observations yield the following result.

Lemma 3.3. *If for every optimal solution y_* of (Q), the optimization problem*

$$(Q_*) \quad \min_{x \in \mathcal{X}} \frac{y_*^T f(x)}{y_*^T g(x)}$$

has a unique optimal solution, then the new algorithm converges superlinearly.

Proof. It follows easily, from (19), (20) and Lemma 3.2, that \limsup equals zero. \square

In order to guarantee the condition in the above lemma we need to introduce the following subset of quasiconvex functions [1].

Definition 3.1. The function $q: \mathcal{X} \rightarrow \mathbb{R}$ is called *strictly quasiconvex* if for each $x_1, x_2 \in \mathcal{X}$ with $x_1 \neq x_2$

$$q(\lambda x_1 + (1 - \lambda)x_2) < \max\{q(x_1), q(x_2)\},$$

for every $0 < \lambda < 1$.

Observe by [1, Proposition 3.29] that $\min_{x \in \mathcal{X}} q(x)$ has a unique optimal solution if $q: \mathcal{X} \rightarrow \mathbb{R}$ is strictly quasiconvex. Hence we have proved the following result.

Corollary 3.1. *If for every $y \in \Sigma$ the function $q: \mathcal{X} \rightarrow \mathbb{R}$ given by $q(x) = y^T f(x)/y^T g(x)$ is strictly quasiconvex, then the proposed algorithm converges superlinearly.*

The next lemma establishes sufficient conditions to achieve strict quasiconvexity.

Lemma 3.4. *If $f: \mathcal{X} \rightarrow \mathbb{R}^m$ is positive, strictly convex and $g: \mathcal{X} \rightarrow \mathbb{R}^m$ is positive, concave or $f: \mathcal{X} \rightarrow \mathbb{R}^m$ is positive, convex and $g: \mathcal{X} \rightarrow \mathbb{R}^m$ is positive, strictly concave, then for every $y \in \Sigma$ the function $q: \mathcal{X} \rightarrow \mathbb{R}$ given by $q(x) = y^T f(x)/y^T g(x)$ is strictly quasiconvex.*

Proof. Clearly for every $x_1, x_2 \in \mathcal{X}$ with $x_1 \neq x_2$ and $0 < \lambda < 1$ it follows that

$$\begin{aligned} q(\lambda x_1 + (1 - \lambda)x_2) &< \frac{\lambda y^T f(x_1) + (1 - \lambda)y^T f(x_2)}{\lambda y^T g(x_1) + (1 - \lambda)y^T g(x_2)} \\ &= \frac{\lambda y^T f(x_1)/(y^T g(x_1))y^T g(x_1) + (1 - \lambda)y^T f(x_2)/(y^T g(x_2))y^T g(x_2)}{\lambda y^T g(x_1) + (1 - \lambda)y^T g(x_2)} \\ &\leq \max \left\{ \frac{y^T f(x_1)}{y^T g(x_1)}, \frac{y^T f(x_2)}{y^T g(x_2)} \right\} \\ &= \max\{q(x_1), q(x_2)\}. \end{aligned} \tag{21}$$

which completes the proof. \square

Note that if $f : \mathcal{X} \rightarrow \mathbb{R}^m$ is strictly convex and $g : \mathcal{X} \rightarrow \mathbb{R}^m$ is positive and affine, then the function $q : \mathcal{X} \rightarrow \mathbb{R}$ given by $q(x) = y^\top f(x) / y^\top g(x)$ is also strictly quasiconvex.

We will now discuss in more detail Step 1 of our algorithm. At the k th iteration, we have to solve in Step 1 the optimization problem where $\mu := c(y_k)$

$$(Q_\mu) \quad \max_{y \in \Sigma} \min_{x \in \mathcal{X}} \{y^\top (f(x) - \mu g(x))\}.$$

Unfortunately, solving this problem may take a lot of time, and this will influence the practical applicability of the new method. On the other hand, when applying the Dinkelbach-type algorithm we need to solve in each step the optimization problem (P_μ) which seems to be easier. However, under some reasonable assumptions it is possible to relate an optimal solution x_{k+1} of $(P_{c(y_k)})$ to an optimal solution y_{k+1} of $(Q_{c(y_k)})$. To derive this relation, we assume that the nonempty compact convex set \mathcal{X} is given by

$$\mathcal{X} := \{x \in \mathbb{R}^n : d_l^\top x \leq \gamma_l, \quad l = 1, \dots, r, \quad p_j(x) \leq 0, \quad j = 1, \dots, s\},$$

where $d_l \in \mathbb{R}^n$, $\gamma_l \in \mathbb{R}$, $l = 1, \dots, r$, and $p_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 1, \dots, s$ are convex and differentiable functions.

Under our standard assumptions it is clear that problem $(P_{c(y_k)})$ is equivalent to the following convex programming problem:

$$(P_k) \quad \min \quad t$$

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

$$p_j(x) \leq 0, \quad \forall j = 1, \dots, s,$$

$$d_l^\top x - \gamma_l \leq 0, \quad \forall l = 1, \dots, r,$$

with $q_i(x) := f_i(x) - c(y_k)g_i(x)$, $i = 1, \dots, m$. To continue our analysis we assume, besides the standard assumptions on the functions f_i and g_i , $i = 1, \dots, m$, that these functions are differentiable.

Let x_{k+1} and t_{k+1} be an optimal solution of the above problem, and define $I' := \{1 \leq i \leq m : q_i(x_{k+1}) = t_{k+1}\}$, $J' := \{1 \leq j \leq s : p_j(x_{k+1}) = 0\}$ and $L' := \{1 \leq l \leq r : d_l^\top x_{k+1} = \gamma_l\}$. If some constraint qualification is satisfied, [4], then the Karush–Kuhn–Tucker conditions ensure the existence of nonnegative scalars u_i , $i \in I'$, v_j , $j \in J'$, and ξ_l , $l \in L'$, satisfying

$$\sum_{i \in I'} u_i \nabla q_i(x_{k+1}) + \sum_{j \in J'} v_j \nabla p_j(x_{k+1}) + \sum_{l \in L'} \xi_l d_l = 0, \tag{22}$$

$$\sum_{i \in I'} u_i = 1, \tag{23}$$

$$(u_{I'}, v_{J'}, \xi_{L'}) \geq 0. \tag{24}$$

Observe, due to the special structure of the problem (P_k) , we only have to impose a constraint qualification over the set of constraints \mathcal{X} . Moreover, since the above problem is convex, Slater’s condition, i.e., there exist some $x_0 \in \mathbb{R}^n$ with $d_l^\top x_0 \leq \gamma_l$, $l = 1, \dots, r$, and $p_j(x_0) < 0$ for every $j = 1, \dots, s$, is such an example of a constraint qualification [16]. Additional constraint qualifications can be found in [4,6,32].

Notice that the set I' is nonempty due to the optimality of $(\mathbf{x}_{k+1}, t_{k+1})$. It is now possible to relate the scalars $u_i, i \in I'$ to an optimal solution of $(Q_{c(y_k)})$.

Lemma 3.5. *If some constraint qualification holds on \mathcal{X} , then an optimal solution $\hat{\mathbf{y}}$ of $(Q_{c(y_k)})$ is given by*

$$\hat{y}_i = \begin{cases} 0, & \text{if } i \notin I', \\ u_i, & \text{if } i \in I', \end{cases}$$

where u_i solves the system (22)–(24).

Proof. From (23) and (24) it follows that $\hat{\mathbf{y}}$ belongs to Σ . Moreover, by the definition of I' we obtain that $\sum_{i \in I'} \hat{y}_i q_i(\mathbf{x}_{k+1}) = t_{k+1}$. This yields by (7) that

$$\sum_{i \in I'} \hat{y}_i q_i(\mathbf{x}_{k+1}) = \min_{\mathbf{x} \in \mathcal{X}} \max_{i \in I} \{f_i(\mathbf{x}) - c(y_k) g_i(\mathbf{x})\} = \max_{\mathbf{y} \in \Sigma} F(\mathbf{y}, c(y_k)).$$

It is left to show that $\hat{\mathbf{y}}$ is an optimal solution of $\max_{\mathbf{y} \in \Sigma} F(\mathbf{y}, c(y_k))$. Since $\min_{\mathbf{x} \in \mathcal{X}} \hat{\mathbf{y}}^T \mathbf{q}(\mathbf{x})$ is a convex optimization problem, the Karush–Kuhn–Tucker conditions are sufficient [16]. Clearly by the definition of $\hat{\mathbf{y}}$ and (22)–(24) the vector \mathbf{x}_{k+1} satisfies these conditions, and thus \mathbf{x}_{k+1} is an optimal solution of $\min_{\mathbf{x} \in \mathcal{X}} \hat{\mathbf{y}}^T \mathbf{q}(\mathbf{x})$. Hence, $\hat{\mathbf{y}} \in \Sigma$ satisfies

$$\max_{\mathbf{y} \in \Sigma} F(\mathbf{y}, c(y_k)) = \sum_{i \in I'} \hat{y}_i q_i(\mathbf{x}_{k+1}) = F(\hat{\mathbf{y}}, c(y_k)).$$

and so $\hat{\mathbf{y}}$ solves $(Q_{c(y_k)})$. \square

Due to numerical errors the system given by (22)–(24) may appear to be “inconsistent”. To solve this problem, observe first that this linear system can be rewritten as follows

$$A\mathbf{u} + B_1\mathbf{v} + B_2\xi = \mathbf{0}, \quad \mathbf{u} \in \Sigma, \mathbf{v}, \xi \geq \mathbf{0}.$$

Letting $E := [AB_1B_2]^T[AB_1B_2]$ and $\mathbf{w} = (\mathbf{u}, \mathbf{v}, \xi)$, it follows that solving this linear system corresponds to finding a nonnegative vector $\mathbf{w} \in \mathbb{R}^v$ where $v := |I'| + |J'| + |L'|$, with the smallest ellipsoidal norm $\sqrt{\mathbf{w}^T E \mathbf{w}}$ under the constraint that its first $|I'|$ components belong to the unit simplex, or equivalently:

$$\min \frac{1}{2} \mathbf{w}^T E \mathbf{w} \tag{25}$$

$$\mathbf{u} \in \Sigma, \mathbf{v}, \xi \geq \mathbf{0}. \tag{26}$$

Clearly, in the presence of no numerical errors the optimal objective value of this problem is zero.

In order to conclude the discussion of the “dual” algorithm it is important to consider a stopping rule for Step 2. Due to Lemma 3.5 and (7) a stopping rule can be derived similarly as for the Dinkelbach-type algorithm. In fact, from [11, Proposition 2.2] it follows that stopping the “dual” algorithm whenever $F(\mu_k) \leq \varepsilon g(\mathbf{x}_k)$, with $\mu_k := c(y_k)$ the current iteration point and $\mathbf{x}_k \in \mathcal{X}$ an optimal solution of (P_{μ_k}) and $\underline{g}(\mathbf{x}) := \min_{i \in I} g_i(\mathbf{x})$, yields $\vartheta(Q) - c(y_k) \leq \varepsilon$. Observe that, for the Dinkelbach-type

algorithm the stopping rule corresponds to $F(\mu_k) \leq \varepsilon \delta$ with $\delta := \min_{x \in \mathcal{X}} \min_{i \in I} g_i(x)$. Clearly, the same rule can also be used for the “dual” algorithm.

It is important to mention that in the special case of generalized linear fractional programming, i.e., the functions f_i, g_i involved are linear and the set \mathcal{X} is a nonempty polytope, Step 1 reduces to solving a linear programming problem, see [2].

4. A type-2 version of the new algorithm

Following the same strategy used to derive the Dinkelbach-type-2 algorithm we will propose a variant of the new algorithm introduced in the previous section. Before presenting this variant we introduce for $x_k \in \mathcal{X}$ the vector-valued functions $f^{(k)}, g^{(k)}$ given by $f_i^{(k)}(x) := f_i(x)/g_i(x_k)$ and $g_i^{(k)}(x) := g_i(x)/g_i(x_k)$. We can now define the optimization problem

$$(Q^{(k)}) \quad \max_{y \in \Sigma} c^{(k)}(y),$$

with

$$c^{(k)}(y) := \min_{x \in \mathcal{X}} \frac{y^T f^{(k)}(x)}{y^T g^{(k)}(x)}.$$

Denote by $y_*^{(k)}$ an optimal solution of the optimization problem $(Q^{(k)})$. By similar arguments as used in the previous section we obtain

$$\begin{aligned} c^{(k)}(y_*^{(k)}) &= \max_{y \in \Sigma} c^{(k)}(y) = \max_{y \in \Sigma} \left\{ \min_{x \in \mathcal{X}} \frac{y^T f^{(k)}(x)}{y^T g^{(k)}(x)} \right\} \\ &= \min_{x \in \mathcal{X}} \left\{ \max_{y \in \Sigma} \frac{y^T f^{(k)}(x)}{y^T g^{(k)}(x)} \right\} = \min_{x \in \mathcal{X}} \left\{ \max_{i \in I} \frac{f_i^{(k)}(x)}{g_i^{(k)}(x)} \right\} \\ &= \min_{x \in \mathcal{X}} \left\{ \max_{i \in I} \frac{f_i(x)}{g_i(x)} \right\} = \mu^* = c(y_*), \end{aligned} \tag{27}$$

and so for every $y \in \Sigma$ it follows that

$$c^{(k)}(y) \leq c^{(k)}(y_*^{(k)}) = \mu^*.$$

Similar to the approach used in the previous section consider the parametric problem associated with $(Q^{(k)})$ given by

$$(Q_\mu^{(k)}) \quad \max_{y \in \Sigma} F^{(k)}(y, \mu),$$

with

$$F^{(k)}(y, \mu) := \min_{x \in \mathcal{X}} \{y^T (f^{(k)}(x) - \mu g^{(k)}(x))\}.$$

Let also y_k be an optimal solution of $(Q_\mu^{(k)})$ with $\mu = c^{(k-1)}(y_{k-1})$, i.e.,

$$y_k := \operatorname{argmax}_{y \in \Sigma} F^{(k)}(y, \mu).$$

In order to simplify the notation we will use, whenever there is no danger of confusion, $c'(y_k)$ instead of $c^{(k)}(y_k)$ and $c'(y_*^{(k)})$ instead of $c^{(k)}(y_*^{(k)})$.

The equivalence relation established in (7) can also be derived for this case due to the convexity/concavity assumptions of the functions $f^{(k)}, g^{(k)}$ for all $k \geq 0$. To be more precise, it follows by Von Neumann's min-max theorem that

$$\begin{aligned}
 F^{(k)}(y_k, c'(y_{k-1})) &= \max_{y \in \Sigma} \left\{ \min_{x \in \mathcal{X}} \left\{ y^T (f^{(k)}(x) - c'(y_{k-1})g^{(k)}(x)) \right\} \right\} \\
 &= \min_{x \in \mathcal{X}} \left\{ \max_{y \in \Sigma} \left\{ y^T (f^{(k)}(x) - c'(y_{k-1})g^{(k)}(x)) \right\} \right\} \\
 &= \min_{x \in \mathcal{X}} \left\{ \max_{i \in I} \left\{ f_i^{(k)}(x) - c'(y_{k-1})g_i^{(k)}(x) \right\} \right\} \\
 &= F^{(k)}(c'(y_{k-1})), \tag{28}
 \end{aligned}$$

with $F^{(k)}: \mathbb{R} \rightarrow \mathbb{R}$ the parametric function used in the Dinkelbach-type-2 algorithm. However, while in the Dinkelbach-type-2 algorithm x_k is an optimal solution of $(P_{\mu}^{(k-1)})$, the vector x_k in this variant must be an optimal solution of the fractional programming problem

$$c'(y_{k-1}) := \min_{x \in \mathcal{X}} \frac{y_{k-1}^T f^{(k-1)}(x)}{y_{k-1}^T g^{(k-1)}(x)}.$$

Observe that due to Lemma 2.1 this implies that $c'(y_k)$ is the root of the parametric equation $F^{(k)}(y_k, \mu) = 0$.

Assuming that the ‘‘scaling’’ points x_k are obtained as described this variant of the ‘‘dual’’ algorithm requires, as already observed, solving the parametric problem $(Q_{\mu}^{(k)})$ with $\mu = c^{(k-1)}(y_{k-1})$ to obtain the next iteration point y_k . The modified algorithm is described by the following procedure.

Algorithm 4.1 (‘‘Dual’’-2 algorithm).

Step 0'. Take $y_0 \in \Sigma$ and $x_0 \in \mathcal{X}^o$;

 Compute $c'(y_0)$ and let x_1 be an optimal solution of $c'(y_0)$;

 Let $k := 1$;

Step 1'. Determine $y_k := \operatorname{argmax}_{y \in \Sigma} F^{(k)}(y, c'(y_{k-1}))$;

Step 2'. If $F^{(k)}(y_k, c'(y_{k-1})) = 0$

Then y_k is an optimal solution of $(Q^{(k)})$ with value $c'(y_{k-1})$ and **Stop**.

Else Goto Step 3';

Step 3'. Compute $c'(y_k)$ and let x_{k+1} be an optimal solution of $c'(y_k)$;

 Let $k := k + 1$ and **GoTo** Step 1'.

As before we denote by k^* the number of times that Step 1' of the ‘‘dual’’-2 algorithm was started. Clearly, if k^* equals $+\infty$ the ‘‘dual’’-2 algorithm does not stop while for k^* finite it follows that $F^{(k^*)}(y_{k^*}, c'(y_{k^*-1})) = 0$. Before discussing the convergence of the sequence $c'(y_{k-1}), k \geq 0$, we introduce as in (10) the set

$$\mathcal{X}_y^{(k)}(\mu) := \left\{ x \in \mathcal{X} : y^T (f^{(k)}(x) - \mu g^{(k)}(x)) = F_y^{(k)}(\mu) \right\},$$

with $F_y^{(k)}(\mu) := F^{(k)}(y, \mu)$. If $\varphi_k : \Sigma \rightarrow \mathbb{R}^m$ is given by

$$\varphi_k(y) := \left(\frac{y_1}{g_1(x_k)}, \dots, \frac{y_m}{g_m(x_k)} \right)^T,$$

then clearly $\mathcal{X}_y^{(k)}(\mu) = \mathcal{X}_{\varphi(y)}(\mu)$ and $c^{(k)}(y) = c(\varphi_k(y))$. Moreover, if $\psi_k : \Omega_k \rightarrow \mathbb{R}^m$ with $\Omega_k := \{y \in \mathbb{R}_+^m : y^T g(x_k) > 0\}$ is given by

$$\psi_k(y) := \left(\frac{y_1 g_1(x_k)}{y^T g(x_k)}, \dots, \frac{y_m g_m(x_k)}{y^T g(x_k)} \right)^T,$$

then it is easy to show that y_* is an optimal solution of (Q) if and only if $y_* \in \Sigma$ and $\psi_k(y_*)$ is an optimal solution of $(Q^{(k)})$. This observation implies that there exists for any optimal solution $y_*^{(k)}$ of $(Q^{(k)})$ some optimal solution y_* of (Q) satisfying $y_*^{(k)} = \psi_k(y_*)$. Letting

$$\Delta_{k-1}^{(k)}(y) := \max\{y^T g^{(k)}(x) : x \in \mathcal{X}_y^{(k)}(c'(y_{k-1}))\},$$

it follows that

$$\begin{aligned} \Delta_{k-1}^{(k)}(y_*^{(k)}) &= \max\{\psi_k(y_*)^T g^{(k)}(x) : x \in \mathcal{X}_{\psi_k(y_*)}^{(k)}(c'(y_{k-1}))\} \\ &= \frac{1}{y_*^T g(x_k)} \max\{y_*^T g(x) : x \in \mathcal{X}_{\varphi_k(\psi_k(y_*))}^{(k)}(c'(y_{k-1}))\} \\ &= \frac{1}{y_*^T g(x_k)} \max\{y_*^T g(x) : x \in \mathcal{X}_{y_*}^{(k)}(c'(y_{k-1}))\}, \end{aligned} \tag{29}$$

whenever $y_*^{(k)} = \psi_k(y_*)$. Also, if

$$\delta_k^{(k)} := \min\{y_k^T g^{(k)}(x) : x \in \mathcal{X}_{y_k}^{(k)}(c'(y_k))\},$$

we obtain that

$$\delta_k^{(k)} = \min\{\varphi_k(y_k)^T g(x) : x \in \mathcal{X}_{\varphi_k(y_k)}^{(k)}(c(\varphi_k(y_k)))\}.$$

Observe now the function $F_y^{(k)} : \mathbb{R} \rightarrow \mathbb{R}$, for k and y fixed, has the same properties as the function F_y considered in the previous section and so as in Lemma 3.2 we obtain

$$\partial(-F_y^{(k)})(\mu) = \left[\min_{x \in \mathcal{X}_y^{(k)}(\mu)} \{y^T g^{(k)}(x)\}, \max_{x \in \mathcal{X}_y^{(k)}(\mu)} \{y^T g^{(k)}(x)\} \right]. \tag{30}$$

Clearly, by this result we have that

$$\Delta_{k-1}^{(k)}(y) \in \partial(-F_y^{(k)})(c'(y_{k-1})), \tag{31}$$

$$\delta_k^{(k)} \in \partial(-F_{y_k}^{(k)})(c'(y_k)) = \partial(-F_{\varphi_k(y_k)})(c(\varphi_k(y_k))). \tag{32}$$

It is now possible to prove the following result. Observe the proof of this result resembles the proof of Theorem 3.1.

Theorem 4.1. *The sequence $y_k, 0 \leq k < k^*$, generated by the “dual”-2 algorithm does not contain optimal solutions of (Q) and the corresponding function values $c'(y_k), 0 \leq k < k^*$, are strictly increasing. Moreover, if k^* is finite, it follows that $c'(y_{k^*}) = \mu^*$, while for k^* equals $+\infty$, $\lim_{k \uparrow \infty} c'(y_k) = \mu^*$. In particular, the inequality*

$$0 \leq \mu^* - c'(y_k) \leq \left(1 - \frac{\Delta_{k-1}^{(k)}(y_*^{(k)})}{\delta_k^{(k)}}\right) (\mu^* - c'(y_{k-1})) \tag{33}$$

holds for every $k \geq 1$ with $y_*^{(k)}$ an optimal solution of $(Q^{(k)})$.

Proof. The proof of the first part of this result is similar to the first part of the proof of Theorem 3.1 and thus it is omitted.

To verify that $\lim_{k \uparrow \infty} c'(y_k)$ exists for $k^* = +\infty$ and that (33) holds we observe the following. Since for $k^* = +\infty$, the sequence $\{c'(y_k)\}_{k \geq 0}$ is strictly increasing it follows that $\lim_{k \uparrow \infty} c'(y_k)$ exists. Moreover, if $y_*^{(k)}$ is an optimal solution of $(Q^{(k)})$ we obtain due to (31) and $c'(y_*^{(k)}) = \mu^*$ that

$$F^{(k)}(y_*^{(k)}, \mu^*) - F^{(k)}(y_*^{(k)}, c'(y_{k-1})) \leq -(\mu^* - c'(y_{k-1})) \Delta_{k-1}^{(k)}(y_*^{(k)}).$$

Using $F^{(k)}(y_*^{(k)}, \mu^*) = 0$ this implies that

$$\begin{aligned} F^{(k)}(y_k, c'(y_{k-1})) &= \max_{y \in \Sigma} F^{(k)}(y, c'(y_{k-1})) \geq F^{(k)}(y_*^{(k)}, c'(y_{k-1})) \\ &\geq (\mu^* - c'(y_{k-1})) \Delta_{k-1}^{(k)}(y_*^{(k)}). \end{aligned} \tag{34}$$

Also, by (32) we obtain that

$$\begin{aligned} F^{(k)}(y_k, c'(y_{k-1})) &= F^{(k)}(y_k, c'(y_{k-1})) - F^{(k)}(y_k, c'(y_k)) \\ &\leq (c'(y_k) - c'(y_{k-1})) \delta_k^{(k)}. \end{aligned} \tag{35}$$

Combining the above inequality with (34) yields

$$\delta_k^{(k)}(c'(y_k) - c'(y_{k-1})) \geq (\mu^* - c'(y_{k-1})) \Delta_{k-1}^{(k)}(y_*^{(k)}). \tag{36}$$

By this inequality it follows that $\lim_{k \uparrow \infty} c'(y_k) = \mu^*$ and (33) is an easy consequence. \square

As for the “dual” algorithm discussed in Section 3 it is important to investigate under which conditions this “dual”-2 algorithm has superlinear convergence. As we will next show the sufficient condition established in Lemma 3.3 for the “dual” algorithm yields for this variant the same convergence rate result.

Lemma 4.1. *If for every optimal solution y_* of (Q) the optimization problem (Q_*) has a unique optimal solution, then the “dual”-2 algorithm converges superlinearly.*

Proof. Let $\delta_x^{(\infty)} := \limsup_{k \uparrow \infty} \delta_k^{(k)}$. Since (y_k, x_k) belongs to the compact set $\Sigma \times \mathcal{X}$ one can find a subsequence $\mathcal{N} \subseteq \mathbb{N}$ such that

$$\lim_{k \in \mathcal{N}, k \uparrow \infty} \delta_k^{(k)} := \delta_x^{(\infty)}, \quad \lim_{k \in \mathcal{N}, k \uparrow \infty} y_k := y_x \in \Sigma, \quad \lim_{k \in \mathcal{N}, k \uparrow \infty} x_k := x_x \in \mathcal{X},$$

and so it follows that $\lim_{k \in \mathcal{K}, k \uparrow \infty} \varphi_k(\mathbf{y}_k) = \nu_x$ with $\nu_{x_i} := y_{x_i}/g_i(\mathbf{x}_x)$ for every $i = 1, \dots, m$. By Theorem 4.1 we know that the sequence $c'(\mathbf{y}_k) = c(\varphi_k(\mathbf{y}_k))$, $k \geq 0$ converges to μ^* and hence $\nu_x(\nu_x^\top \mathbf{e})^{-1}$ is an optimal solution of (Q), where $\mathbf{e} = (1, \dots, 1)^\top$. Take now for every $k \geq 0$ the vector $\psi_k(\nu_x)$. This vector solves $(Q^{(k)})$ and by (29)

$$\Delta_{k-1}^{(k)}(\psi_k(\nu_x)) = \frac{1}{\nu_x^\top \mathbf{g}(\mathbf{x}_k)} \max \left\{ \nu_x^\top \mathbf{g}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}_{\nu_x}(c'(\mathbf{y}_{k-1})) \right\}.$$

Since $c'(\mathbf{y}_{k-1})$ is increasing we obtain by a similar argument as used in the previous section to derive the superlinear convergence rate of the ‘‘dual’’ algorithm that as $k \uparrow \infty$,

$$\max \left\{ \nu_x^\top \mathbf{g}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}_{\nu_x}(c'(\mathbf{y}_{k-1})) \right\}$$

converges to

$$\min \left\{ \nu_x^\top \mathbf{g}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}_{\nu_x}(\mu^*) \right\}.$$

Due to $\lim_{k \in \mathcal{K}, k \uparrow \infty} \mathbf{x}_k = \mathbf{x}_x$ this yields by the definition of ν_x that

$$\lim_{k \in \mathcal{K}, k \uparrow \infty} \nu_x^\top \mathbf{g}(\mathbf{x}_k) = 1$$

and hence

$$\lim_{k \in \mathcal{K}, k \uparrow \infty} \Delta_{k-1}^{(k)}(\psi_k(\nu_x)) = \min \left\{ \nu_x^\top \mathbf{g}(\mathbf{x}) : \mathbf{x} \in \mathcal{X}_{\nu_x}(\mu^*) \right\}. \tag{37}$$

Moreover, since by (32) $\delta_k^{(k)} \in \partial(-F_{\varphi_k(\mathbf{y}_k)})(c(\varphi_k(\mathbf{y}_k)))$, we obtain by the upper semicontinuity of the mapping $\mathbf{y} \mapsto \partial(-F_{\mathbf{y}})(c(\mathbf{y}))$ that

$$\delta_x^{(\infty)} \in \partial(-F_{\nu_x})(\mu^*). \tag{38}$$

Using now Theorem 4.1, it follows that

$$\mu^* - c'(\mathbf{y}_k) \leq \left(1 - \frac{\Delta_{k-1}^{(k)}(\psi_k(\nu_x))}{\delta_k^{(k)}} \right) (\mu^* - c'(\mathbf{y}_{k-1})).$$

Combining the above inequality with (37) and (38) and the unicity assumption of the optimal solution of (Q_*) yields the desired result. \square

Similar to the ‘‘dual’’ algorithm, the ‘‘difficult’’ problem in Step 1’ can also be solved as described in the previous section.

Finally, it is left to consider stopping rules. Since the scaled algorithms consider a scaled parametric function the stopping rule has to be adjusted accordingly. For instance, in the Dinkelbach-type-2 algorithm Ferland and Potvin [14] use as stopping rule $\varepsilon/g_{i_k}(\mathbf{x}_k)$, where $i_k \in I$ is the index where the maximum is attained in problem $(P_{\mu_k}^{(k-1)})$ in Step 2’. Using the approach described at the end of the previous section, a similar rule can be derived for the ‘‘dual’’-2 algorithm.

5. Numerical results

In order to test the efficiency of the “dual” method we compared it with the extension of Dinkelbach’s method to several ratios.

In the test problems considered the numerator of the ratios are quadratic functions $f_i(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T\mathbf{H}_i\mathbf{x} + \mathbf{a}_i^T\mathbf{x} + b_i$, and the denominator are linear functions, $g_i(\mathbf{x}) := \mathbf{c}_i^T\mathbf{x} + d_i$. The quadratic functions, f_i , are generated in the following way.

- In the linear term each element of the vector \mathbf{a}_i is uniformly drawn from $[-15.0, 45.0]$. Similarly b_i is drawn uniformly from $[-30.0, 0]$;

- The Hessian is defined by $\mathbf{H}_i := \mathbf{L}_i\mathbf{U}_i\mathbf{L}_i^T$ where \mathbf{L}_i is a unit lower triangular matrix with components uniformly drawn from $[-2.5, 2.5]$ and \mathbf{U}_i is a positive diagonal matrix, with elements uniformly drawn from $[0.1, 1.6]$. When a positive semidefinite Hessian is required the first component of the diagonal matrix is set to zero.

The linear functions, g_i , are constructed using a similar procedure: each element of the vector \mathbf{c}_i is uniformly drawn from $[0.0, 10.0]$. Similarly d_i is drawn uniformly from $[1.0, 5.0]$. Finally, the feasible domains considered are the following:

$$\begin{aligned} \mathcal{X}_1 &:= \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{j=1}^n x_j \leq 1, x_j \geq 0, j = 1, \dots, n \right\}, \\ \mathcal{X}_2 &:= \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{j \in J_1} x_j \leq 1, \sum_{j \in J_2} x_j \leq 1, x_j \geq 0, j = 1, \dots, n \right\}, \end{aligned} \tag{39}$$

where $J_1 := \{1 \leq j \leq n : j \text{ is odd}\}$ and $J_2 := \{1 \leq j \leq n : j \text{ is even}\}$.

Both methods were implemented in Sun Pascal, linked to a pair of existing routines written in Sun FORTRAN and ran on a Sun Sparc System 600 using the default double precision (64-bit IEEE floating-point format) real numbers of Sun Pascal and FORTRAN. Both compilers were used with the default compilation options.

For the minimization of the maximum of quadratic functions with linear constraints we used the bundle trust method coded in FORTRAN [20]. In the “dual” type algorithms Step 1 is solved by computing the corresponding minimal ellipsoidal norm problem, see Section 3. The fractional programming problem that occurs in Step 0 and 3 of the algorithm is solved by Dinkelbach’s algorithm [13]. The code used to solve the above quadratic problems is an implementation in FORTRAN of Lemke’s algorithm [23].

In the “dual”-2 algorithms we used in Step 0 $\mathbf{y}_0^T := (1/m, \dots, 1/m)$. In order to have similar starting points in both the algorithms tested we take in Step 0 of the Dinkelbach-type algorithm:

$$\mu_1 := c(\mathbf{y}_0) = \min_{\mathbf{x} \in \mathcal{X}} \frac{\mathbf{y}_0^T \mathbf{f}(\mathbf{x})}{\mathbf{y}_0^T \mathbf{g}(\mathbf{x})}.$$

On the other hand, for the “dual”-2 algorithm we used $\mathbf{x}_0^T := (0, \dots, 0)$. As for the Dinkelbach-type-2 algorithm we used in the initial step:

$$\mu_1 := c^{(0)}(\mathbf{y}_0) = \min_{\mathbf{x} \in \mathcal{X}} \frac{\mathbf{y}_0^T \mathbf{f}^{(0)}(\mathbf{x})}{\mathbf{y}_0^T \mathbf{g}^{(0)}(\mathbf{x})}.$$

Table 1
 \mathcal{L}_1 and strictly quasiconvex ratios

Problem			Dinkelbach			"Dual"					Stat.	
<i>T</i>	<i>n</i>	<i>m</i>	It	%Par	Sec	It	%Fr	%Par	%KS	Sec	%A	%Imp
1	5	5	8	99.5	0.88	3	11.1	85.0	3.9	0.65	32.0	26.5
2	10	5	10	99.9	9.83	3	8.1	91.1	0.6	4.08	64.0	58.5
3	15	5	9	99.5	14.62	3	18.0	79.9	1.9	7.57	44.0	48.2
4	20	5	8	99.9	35.32	3	9.1	90.2	0.7	21.25	64.0	39.8
5	5	10	8	99.1	2.42	4	17.5	79.1	1.9	0.54	22.0	77.7
6	10	10	13	99.6	10.94	4	10.3	88.2	1.1	4.55	36.0	58.4
7	15	10	9	99.7	18.99	3	10.0	88.6	1.2	11.57	36.0	39.1
8	20	10	10	99.7	47.50	3	9.9	89.2	0.7	25.09	34.0	47.2
9	5	15	8	99.3	3.40	3	9.4	90.1	0.5	1.02	13.3	70.1
10	10	15	11	99.4	11.17	3	10.4	88.1	1.3	4.79	29.3	57.1
11	15	15	9	99.6	24.45	3	9.9	88.9	0.9	14.23	28.0	41.8
12	20	15	11	99.7	68.96	3	9.5	89.7	0.7	28.65	34.7	58.5
13	5	20	9	97.1	1.59	4	11.0	85.8	2.3	1.01	15.0	36.7
14	10	20	11	99.2	13.73	4	10.4	88.2	1.2	5.49	21.0	60.0
15	15	20	11	99.5	34.85	4	9.0	89.7	0.9	15.01	24.0	56.9
16	20	20	13	99.6	74.75	3	9.6	89.5	0.7	34.91	31.0	53.3

The tolerance used in Step 2 of the Dinkelbach-type and "dual" algorithms was $\varepsilon := 5 \times 10^{-6}$, see Section 3. For the type-2 variants we considered $\varepsilon \times g_{i_k}(\mathbf{x}_k)$ where $i_k \in I$ is the index where the maximum is attained in problem $(P_{\mu_k}^{(k-1)})$ in Step 2', see Section 4.

In Tables 1–4 we summarize the results of our computational experience comparing the "dual" algorithm with its "primal" counterpart, the Dinkelbach-type algorithm. We

Table 2
 \mathcal{L}_2 and strictly quasiconvex ratios

Problem			Dinkelbach			"Dual"					Stat.	
<i>T</i>	<i>n</i>	<i>m</i>	It	%Par	Sec	It	%Fr	%Par	%KS	Sec	%A	%Imp
1	5	5	7	99.4	2.10	2	12.3	85.4	1.7	0.79	32.0	62.5
2	10	5	11	99.9	12.46	3	9.0	90.1	0.9	5.63	68.0	54.8
3	15	5	9	99.7	21.79	3	11.2	87.9	0.8	11.46	64.0	47.4
4	20	5	9	99.9	41.48	3	9.7	89.7	0.6	21.88	72.0	47.3
5	5	10	11	99.7	2.47	4	12.1	85.4	2.6	0.87	22.0	64.9
6	10	10	12	99.5	11.65	3	8.8	90.3	0.7	5.45	38.0	53.2
7	15	10	12	99.8	37.65	3	8.2	91.2	0.6	16.82	52.0	55.3
8	20	10	10	99.8	59.19	3	9.7	89.7	0.6	31.12	48.0	47.4
9	5	15	9	96.9	1.64	3	15.6	81.3	3.2	0.77	14.7	53.4
10	10	15	11	99.6	15.34	3	7.7	91.6	0.7	6.98	34.7	54.5
11	15	15	9	99.6	29.01	3	8.7	90.4	0.7	17.13	37.3	40.9
12	20	15	10	99.7	70.48	3	9.1	90.2	0.6	35.10	40.0	50.2
13	5	20	8	98.1	1.92	4	12.9	85.4	0.9	1.14	11.0	40.6
14	10	20	11	99.4	13.93	3	8.1	90.7	0.8	7.56	21.0	45.7
15	15	20	13	99.4	37.44	3	8.9	90.1	0.8	16.46	28.0	56.0
16	20	20	11	99.6	80.45	3	7.9	91.5	0.5	38.41	31.0	52.2

Table 3
 \mathcal{P}_1 and semistrictly quasiconvex ratios

Problem			Dinkelbach			"Dual"					Stat.	
<i>T</i>	<i>n</i>	<i>m</i>	It	%Par	Sec.	It	%Fr	%Par	%KS	Sec	%A	%Imp
1	5	5	8	98.0	0.61	5	11.9	80.7	7.5	0.46	60.0	24.5
2	10	5	14	99.5	7.14	3	10.8	87.1	2.1	2.13	56.0	70.2
3	15	5	10	99.7	17.93	3	9.7	88.9	1.2	8.71	64.0	51.4
4	20	5	11	99.8	35.81	3	9.5	89.4	1.0	17.40	52.0	51.4
5	5	10	12	98.2	2.06	4	9.5	85.1	5.4	0.99	30.0	51.7
6	10	10	9	99.6	12.72	4	6.0	92.4	1.3	5.83	26.0	54.1
7	15	10	11	99.6	20.66	3	7.1	91.4	1.3	8.26	34.0	60.0
8	20	10	12	99.7	61.48	3	7.9	91.1	0.9	27.81	52.0	54.8
9	5	15	7	99.8	2.48	4	5.3	93.2	1.5	2.20	13.3	11.2
10	10	15	11	99.6	12.22	3	6.8	91.8	1.1	4.38	26.7	64.1
11	15	15	10	99.6	35.82	3	6.8	92.3	0.9	16.76	30.7	53.2
12	20	15	11	99.6	66.91	3	8.1	90.8	0.9	28.56	33.3	57.3
13	5	20	11	98.1	2.03	4	13.3	81.4	2.2	0.91	13.0	55.0
14	10	20	11	99.0	14.90	4	9.4	88.0	1.8	6.03	19.0	59.5
15	15	20	12	99.3	34.66	3	7.8	90.6	1.3	13.18	29.0	62.0
16	20	20	11	99.6	81.23	4	6.0	93.1	0.6	37.68	24.0	53.6

also present in Tables 5 and 6, for the same test problems, a summary of the computational results obtained with the type-2 variants of these two methods. For each pair (*n*, *m*), where *n* is the number of variables and *m* the number of ratios, five uncorrelated instances of the problem were generated and solved by the four algorithms. Hence, the entries on these tables are averages of the corresponding values. Each class of these test problems is identified by the number in column *T*. The columns under

Table 4
 \mathcal{P}_2 and semistrictly quasiconvex ratios

Problem			Dinkelbach			"Dual"					Stat.	
<i>T</i>	<i>n</i>	<i>m</i>	It	%Par	Sec	It	%Fr	%Par	%KS	Sec	%A	%Imp
1	5	5	8	98.5	0.50	7	19.7	77.6	2.0	0.61	52.0	-21.8
2	10	5	12	99.4	8.62	3	15.0	81.8	3.0	2.40	64.0	72.2
3	15	5	9	99.9	37.11	3	7.3	92.0	0.7	21.03	72.0	43.3
4	20	5	11	99.8	47.47	3	8.7	90.6	0.7	26.15	80.0	44.9
5	5	10	12	99.9	2.98	5	13.6	82.0	4.2	1.07	32.0	64.1
6	10	10	9	99.2	11.61	3	12.5	86.6	0.9	3.89	30.0	66.5
7	15	10	11	99.6	24.42	3	9.0	89.5	1.3	12.07	42.0	50.6
8	20	10	10	99.8	69.70	3	7.7	91.7	0.5	38.87	58.0	44.2
9	5	15	9	99.3	4.62	4	13.6	82.4	2.7	1.13	17.3	75.6
10	10	15	10	99.6	13.76	3	9.5	89.3	0.9	5.65	29.3	59.0
11	15	15	12	99.7	51.50	3	6.5	92.8	0.6	22.18	33.3	56.9
12	20	15	10	99.7	71.50	3	8.3	91.1	0.5	35.82	38.7	49.9
13	5	20	14	98.8	4.15	4	13.6	82.3	2.9	1.13	14.0	72.9
14	10	20	10	99.3	16.38	3	7.9	90.7	1.1	6.82	21.0	58.3
15	15	20	11	99.6	43.66	3	6.7	92.6	0.6	20.32	29.0	53.5
16	20	20	11	99.7	97.39	4	7.2	92.2	0.5	45.49	31.0	53.3

Table 5
Strictly quasiconvex ratios

Problem	\mathcal{R}_1				\mathcal{R}_2			
	Dinkel-2		"Dual-2"		Dinkel-2		"Dual-2"	
	It	Sec	It	Sec	It	Sec	It	Sec
1	5	0.70	3	0.71	6	1.88	3	1.10
2	6	7.64	3	4.66	6	11.93	3	6.45
3	5	11.85	3	8.75	6	18.43	3	11.79
4	5	31.66	3	20.68	5	31.94	3	20.80
5	5	0.61	3	1.45	6	1.23	3	0.76
6	6	6.31	3	4.34	6	6.29	3	5.78
7	6	15.52	3	12.46	6	22.96	3	17.40
8	5	39.92	3	31.60	5	41.42	3	31.12
9	6	2.66	3	2.04	6	1.20	3	1.73
10	6	17.94	3	6.61	6	10.17	3	7.47
11	6	22.16	3	13.84	6	29.13	3	17.12
12	6	49.19	3	33.98	6	53.88	3	34.28
13	5	1.25	4	1.02	5	1.55	3	1.28
14	6	7.49	4	8.09	6	9.58	3	10.54
15	6	23.60	4	17.74	6	23.24	3	15.05
16	6	50.08	3	31.30	6	56.27	3	40.03

Dinkelbach and Dink-2 report the results obtained using the Dinkelbach-type and the Dinkelbach-type-2 algorithm for several ratios. Similarly the columns under "Dual" and "Dual2" report the results obtained using the "dual" and the "dual"-2 algorithm. In the case of "Dual" two extra columns are presented concerning the main steps of this

Table 6
Semistrictly quasiconvex ratios

Problem	\mathcal{R}_1				\mathcal{R}_2			
	Dink-2		"Dual-2"		Dink-2		"Dual-2"	
	It	Sec	It	Sec	It	Sec	It	Sec
1	5	0.47	6	0.85	5	0.47	6	0.83
2	6	3.17	3	2.39	7	5.39	3	3.86
3	6	12.75	3	10.45	6	27.19	4	24.00
4	5	20.15	3	13.90	6	33.61	3	23.12
5	6	1.62	4	1.96	6	1.82	5	1.23
6	5	8.20	3	9.71	6	12.56	3	5.89
7	5	15.05	3	10.28	5	16.64	3	14.13
8	6	34.40	3	30.67	6	44.47	3	37.54
9	6	3.84	4	2.32	6	4.92	4	1.65
10	6	8.36	3	5.33	6	10.63	3	6.20
11	6	26.45	4	19.55	6	37.02	3	21.39
12	6	52.06	3	30.98	6	61.34	3	31.41
13	6	1.16	4	2.00	6	1.63	4	1.49
14	6	11.13	4	6.92	6	13.83	3	9.09
15	6	19.40	3	15.17	6	29.43	3	22.56
16	6	57.39	4	39.34	6	72.43	3	50.53

algorithm. Hence, column %Fr refers to the percentage of the time used to compute the next iteration point, i.e. $c(y_k)$, while column %KS refers to the percentage of the time used to solve the Karush–Kuhn–Tucker system and thus obtaining y_{k+1} , see Section 3. Finally, column %Par refers to the percentage of the time used for optimizing the parametric function $F(\mu)$. Moreover, the column It refers to the number of iterations performed by the corresponding algorithm, while column Sec refers to the average execution time in seconds of the Sun Sparc System 600 workstation measured by the available standard `clock` function of the Sun Pascal library. This measures the elapsed execution time from the start to the end of the corresponding method, excluding input and output operations.

We will start by presenting and analyzing the results using the simpler versions of the two methods, i.e., the “dual” and Dinkelbach-type algorithm. For these cases, the tables also include the column Stat. which contains under column %A the percentage of active ratios of the test problems and under column %Imp the percentage of improvement in

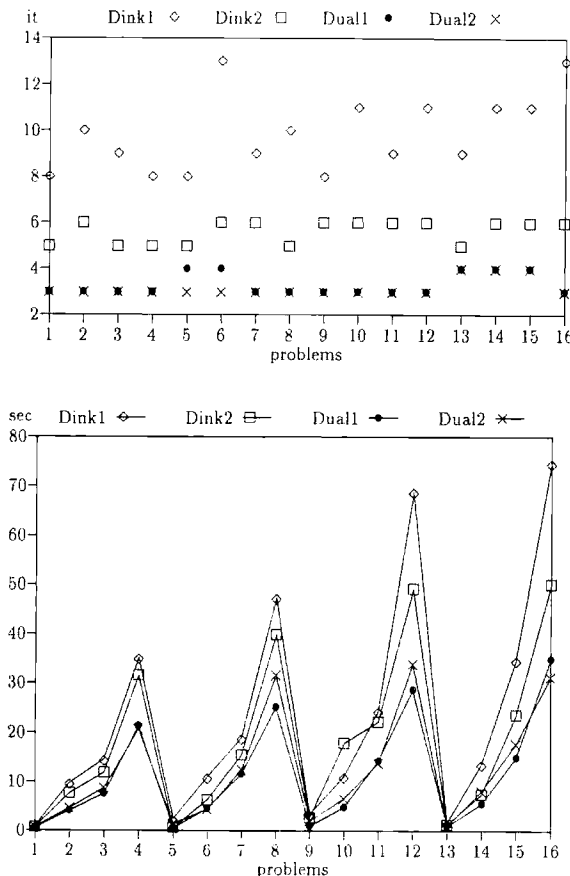


Fig. 2. Strictly quasiconvex ratios and feasible set \mathcal{X}_1 .

total execution time of the “dual” type algorithm over the Dinkelbach-type algorithm, i.e. $(1 - \text{Time}(\text{Dual})/\text{Time}(\text{Din})) \times 100$.

Tables 1 and 2 contain the results obtained for test problems where the quadratic functions f_i are strictly convex. In these cases the convergence rate of the “dual” algorithm is superlinear, see Corollary 3.1.

Although each iteration of the “dual” algorithm is more “expensive” in terms of execution time this extra effort is compensated in the total time used. However, the behavior of the “dual” algorithm seems to be affected by the type of constraints in the feasible set. Indeed when the constraint set is more restrictive (\mathcal{R}_1) the performance of the “dual” algorithm is slightly better. Observe that this phenomenon also occurs for the Dinkelbach-type algorithm.

From the results contained in Tables 1–4 it is clear that the “dual” algorithm has a better performance than the Dinkelbach-type algorithm, not only in terms of the total number of iterations but also in total time, see %Imp.

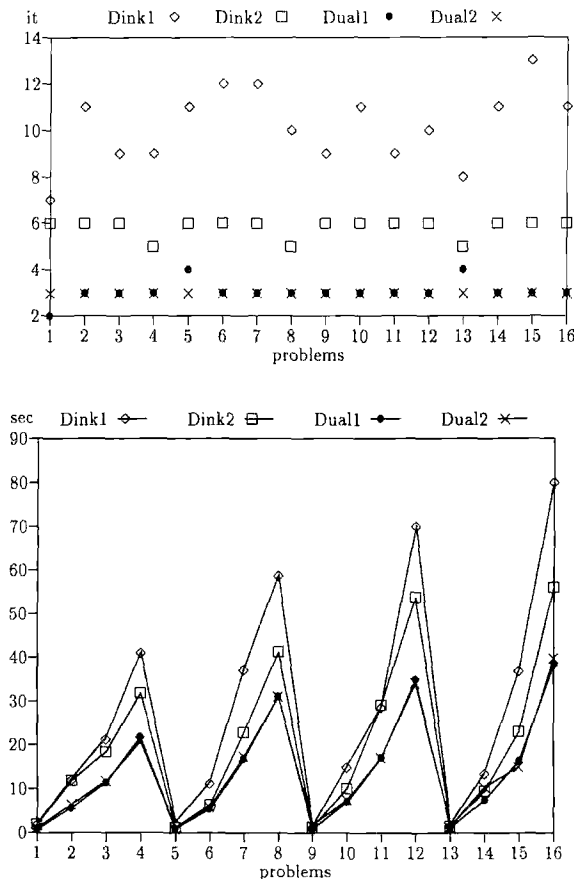


Fig. 3. Strictly quasiconvex ratios and feasible set \mathcal{R} .

Moreover it is also important to remark that even when superlinear convergence cannot be guaranteed, the “dual” algorithm appears to have a better performance than the Dinkelbach-type approach, see Tables 3 and 4.

The results obtained with the type-2 variants of these methods are contained in Tables 5 and 6. Since the total computational time used by the “dual”-2 algorithm appeared to be distributed in a similar way as for its original version these tables are presented in a more condensed form.

From Tables 5 and 6 it appears that in terms of iterations the “dual”-2 algorithm is better than the Dinkelbach-type-2 algorithm. This tendency is also confirmed in terms of total execution time. Moreover, even for the test problems with semistrictly quasiconvex ratios the “dual”-2 algorithm appears to dominate the Dinkelbach-type-2 algorithm.

In order to compare and relate the behavior of the four algorithms we plotted, for the four classes of problems, the number of iterations and total execution time, see Fig. 2–5. In these figures the *x*-coordinates, *problems*, denote the problem type as designated in

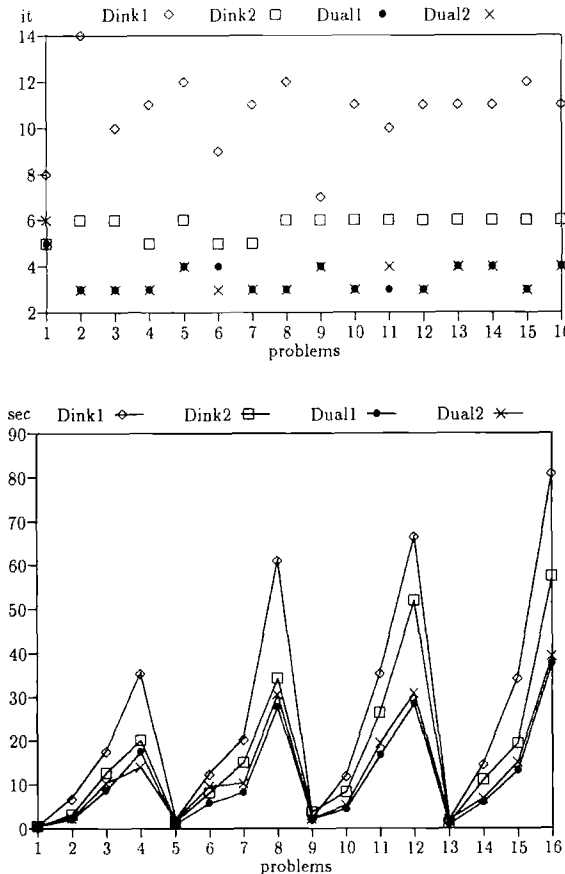


Fig. 4. Semistrictly quasiconvex ratios and feasible set \mathcal{L}_1 .

Tables 1–4, while y -coordinates denote in the first picture the number of iterations it and in the second the total execution time sec .

As expected the Dinkelbach-type-2 algorithm dominates, both in iteration number and execution time, the Dinkelbach-type algorithm. On the other hand, the “dual”-2 algorithm does not produce significant improvements on the behavior of the original “dual” algorithm. This may be explained by the fact that the “dual” algorithm is by itself more robust and “powerful” than its primal counterpart. As Figs. 2–5 show, there are no significant differences between the “dual” type algorithms. In fact, only for the feasible set \mathcal{R}_1 it appears that the “dual” algorithm is slightly better in terms of execution time that its variant. For the case of the feasible set \mathcal{R}_2 these differences are more attenuated. Unexpectedly, the original version of the “dual” algorithm dominates the Dinkelbach-type-2 algorithm both in terms of iterations and execution time.

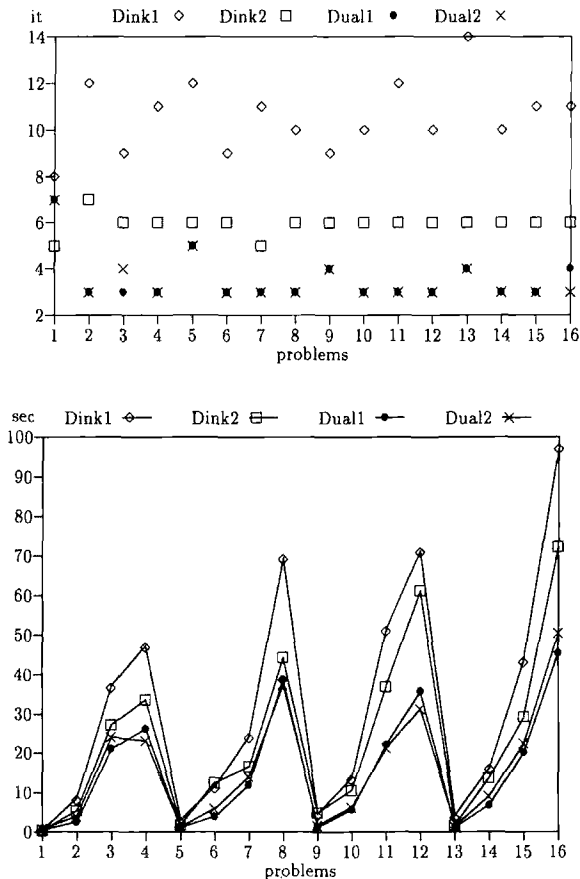


Fig. 5. Semistrictly quasiconvex ratios and feasible set \mathcal{R}_2 .

Figs. 2–5 also show the number of ratios has a decisive influence in the behavior of the “primal” algorithms, Dinkelbach-type and Dinkelbach-type-2 algorithms⁴. This effect is also noticeable for the “dual” type algorithms although at a smaller scale.

Acknowledgements

The authors like to thank J. Outrata, H. Schramm and J. Zowe for making the bundle thrust code available, J. Gromicho for his remarks and suggestions on an earlier version of this paper and the anonymous referee who suggested the scaled version of the “dual” algorithm.

References

- [1] M. Avriel, W.E. Diewert, S. Schaible and I. Zang, *Generalized Convexity*, Mathematical Concepts and Methods in Science and Engineering, Vol. 36 (Plenum, New York, 1988).
- [2] A.I. Barros, *Discrete and Fractional Programming Techniques for Location Models*, Tinbergen Institute Research Series, Vol. 89 (Thesis Publishers, Amsterdam, 1995).
- [3] A.I. Barros and J.B.G. Frenk, “Generalized fractional programming and cutting plane algorithms,” *Journal of Optimization Theory and Applications* 87 (1995) 103–120.
- [4] M.S. Bazaraa, H.D. Sherali and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms* (Wiley, New York, 2nd ed., 1993).
- [5] Y. Benadada, “Approches de résolution du problème de programmation fractionnaire généralisée,” Ph.D. Thesis (Département d’Informatique et de Recherche Opérationnelle, Université de Montréal, 1989).
- [6] A. Ben-Israel, A. Ben-Tal and S. Zlobec, *Optimality in Nonlinear Programming (A Feasible Directions Approach)* (Wiley, New York, 1981).
- [7] J.C. Bernard and J.A. Ferland, “Convergence of interval-type algorithms for generalized fractional programming,” *Mathematical Programming* 43 (1989) 349–364.
- [8] B.D. Craven, *Fractional Programming* (Heldermann, Berlin, 1988).
- [9] J.P. Crouzeix and J.A. Ferland, “Algorithms for generalized fractional programming,” *Mathematical Programming* 52 (1991) 191–207.
- [10] J.P. Crouzeix, J.A. Ferland and S. Schaible, “Duality in generalized linear fractional programming,” *Mathematical Programming* 27 (1983) 342–354.
- [11] J.P. Crouzeix, J.A. Ferland and S. Schaible, “An algorithm for generalized fractional programs,” *Journal of Optimization Theory and Applications* 47 (1985) 35–49.
- [12] J.P. Crouzeix, J.A. Ferland and S. Schaible, “A note on an algorithm for generalized fractional programs,” *Journal of Optimization Theory and Applications* 50 (1986) 183–187.
- [13] W. Dinkelbach, “On nonlinear fractional programming,” *Management Science* 13 (1967) 492–498.
- [14] J.A. Ferland and J.Y. Potvin, “Generalized fractional programming: Algorithms and numerical experimentation,” *European Journal of Operational Research* 20 (1985) 92–101.
- [15] J. Flachs, “Generalized Cheney–Loeb–Dinkelbach-type algorithms,” *Mathematics of Operations Research* 10 (1985) 674–687.
- [16] J.B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms I: Fundamentals*, Vol. 1 (Springer, Berlin, 1993).
- [17] T. Ibaraki, “Parametric approaches to fractional programs,” *Mathematical Programming* 26 (1983) 345–362.

⁴ Recall that the problem types are ordered in the x -axis by the number of ratios.

- [18] T. Ibaraki, H. Ishii, J. Iwase, T. Hasegawa and H. Mine, "Algorithms for quadratic fractional programming problems," *Journal of the Operations Research Society of Japan* 19 (1976) 174–191.
- [19] R. Jagannathan and S. Schaible, "Duality in generalized fractional programming via Farkas' lemma," *Journal of Optimization Theory and Application* 41 (1983) 417–424.
- [20] J. Outrata, H. Schramm and J. Zowe, "Bundle trust methods: Fortran codes for nondifferentiable optimization, User's guide." Technical Report 269 (Mathematisches Institut, Universität Bayreuth, 1991).
- [21] E. Polak, "On the mathematical foundations of nondifferentiable optimization in engineering design," *SIAM Review* 29 (1987) 21–89.
- [22] B.N. Pschenichnyi, *Necessary Conditions for an Extremum* (Marcel Dekker, New York, 1971).
- [23] A. Ravindran, "A computer routine for quadratic and linear programming problems," *Communications of the ACM* 15 (1972) 818.
- [24] A.W. Roberts and D.E. Varberg, *Convex Functions* (Academic Press, New York, 1973).
- [25] R.T. Rockafellar, *Convex Analysis* (Princeton University Press, Princeton, NJ, 1970).
- [26] R.T. Rockafellar, "Generalized subgradients in mathematical programming," in: A. Bachem, M. Grötschel and B. Korte, eds., *Mathematical Programming. The State of the Art* (Springer, Berlin, 1983) ch. 2, pp. 368–390.
- [27] S. Schaible, "Fractional programming," *Zeitschrift für Operations Research* 27 (1983) 39–54.
- [28] S. Schaible and T. Ibaraki, "Fractional programming," (Invited Review), *European Journal of Operational Research* 12 (1983) 325–338.
- [29] N.Z. Shor, *Minimization Methods for Non-differentiable Functions*, Computational Mathematics, Vol. 3 (Springer, Berlin, 1985).
- [30] M. Sion, "On general minimax theorems," *Pacific Journal of Mathematics* 8 (1958) 171–176.
- [31] J. Werner, "Duality in generalized fractional programming," in: K.H. Hoffman, J.B. Hiriart-Urruty, C. Lemaréchal and J. Zowe, eds., *Trends in Mathematical Optimization*, International Series of Numerical Mathematics (Birkhäuser, Basel, 1988) pp. 197–232.
- [32] Z. Zhou, F.S. Mokhtarian and S. Zlobec, "A simple constraint qualification in convex programming," *Mathematical Programming* 61 (1993) 385–397.