Journal of Global Optimization **19:** 83–102, 2001.

Solving the Sum-of-Ratios Problem by an Interior-Point Method

ROLAND W. FREUND[1] and FLORIAN JARRE[2]
[1]*Bell Laboratories, Lucent Technologies, Room 2C–525, 700 Mountain Avenue, Murray Hill, New Jersey 07974–0636, USA (e-mail: freund@research.bell-labs.com)*
[2]*Heinrich-Heine-Universität, Mathematisches Institut, Düsseldorf, Germany (e-mail: jarre@am.uni-duesseldorf.de)*

(Received 11 June 1999; accepted in revised form 21 June 2000)

Abstract. We consider the problem of minimizing the sum of a convex function and of $p \geqslant 1$ fractions subject to convex constraints. The numerators of the fractions are positive convex functions, and the denominators are positive concave functions. Thus, each fraction is quasi-convex. We give a brief discussion of the problem and prove that in spite of its special structure, the problem is $\mathcal{NP}$-complete even when only $p = 1$ fraction is involved. We then show how the problem can be reduced to the minimization of a function of p variables where the function values are given by the solution of certain convex subproblems. Based on this reduction, we propose an algorithm for computing the global minimum of the problem by means of an interior-point method for convex programs.

Key words: Fractional programming, Sum of ratios, Global optimum, Convex subproblem, Interior-point method, $\mathcal{NP}$-completeness, Knapsack problem

1. Introduction

Nonlinear programming problems often involve objective functions that can be expressed in terms of one or several ratios. Exploiting the special structure of such *fractional programs* has been the subject of extensive studies in the last few decades. For an overview of fractional programming, we refer the reader to Schaible (1995) and the references given therein.

Fractional programs with only a single ratio or a maximum of finitely many ratios are fairly well understood. Under suitable conditions, these problems still satisfy some form of generalized convexity, which can be exploited in algorithms for the numerical solution of such problems. For example, there are polynomial-time interior-point methods for classes of such problems (see Freund and Jarre, 1994, 1995; Nemirovskii, 1996).

On the other hand, fractional programs with sums of ratios are much more difficult and not as well understood (see Schaible, 1995, 1996). Such problems possess some form of generalized convexity only in special cases, such as the ones discussed in Schaible (1984) and Hirche (1985), and in general, they have multiple maxima and minima. Algorithms for classes of sum-of-ratios problems are described in Cambini et al. (1989), Chen et al. (1998), Falk and Palocsay (1992), Konno and Kuno (1990), Konno and Yamashita (1998), Ritter (1967), and in the review article (Schaible, 1996). However, most of these algorithms are for the optimization of linear ratios subject to linear constraints. The purpose of this paper is to present a suitable interior-point approach for the solution of much more general problems with convex-concave ratios and convex constraints. Our approach is based on approximating the sum-of-ratios problem by a sequence of convex minimization problems. For such convex problems, interior-point methods have become the methods of choice, both from the point of view of theoretical complexity and of practical efficiency. By using a simple warm-start strategy, the cost for solving the individual convex subproblems can be reduced to very few iterations. Finally, the interior-point method provides certain dual information needed for the overall approach.

More precisely, we consider the problem of minimizing or maximizing the sum of a single function and of $p \geqslant 1$ ratios subject to convex constraints, and we explore the use of interior-point methods for the solution of such problems. More precisely, we study problems of the form

$$\text{minimize} \quad h(x) + \sum_{j=1}^{p} \frac{f_j(x)}{g_j(x)} \qquad \text{subject to} \quad x \in \mathcal{S}, \tag{1}$$

and

$$\text{maximize} \quad h(x) + \sum_{j=1}^{p} \frac{f_j(x)}{g_j(x)} \qquad \text{subject to} \quad x \in \mathcal{S}. \tag{2}$$

Here and in the sequel, we make the following assumptions.

ASSUMPTION 1. *$\mathcal{S} \subset \mathbb{R}^n$ is a compact convex set such that $f_j(x) \geqslant 0$ and $g_j(x) > 0$ for all $j = 1, 2, \ldots, p$ and all $x \in \mathcal{S}$. For the minimization problem* (1)*, the functions h and $f_1, f_2, \ldots, f_p$ are convex and the functions $g_1, g_2, \ldots, g_p$ are concave. For the maximization problem* (2)*, the functions h and $f_1, f_2, \ldots, f_p$ are concave and the functions $g_1, g_2, \ldots, g_p$ are convex.*

For simplicity, from now on we restrict ourselves to minimization problems (1). The results and algorithms for (1) in this paper can easily be converted to maximization problems (2) by simply exchanging ‘min’ and ‘max’, ‘convex’ and ‘concave’, and ‘$\leqslant$’ and ‘$\geqslant$’.

In Section 2, we first discuss the simplest case, namely the sum of a convex function and only $p = 1$ ratio. We show that this problem is $\mathcal{NP}$-complete and propose a method for finding the global minimizer. In Section 3, the method is generalized to the case $p \geqslant 2$. In Section 4, we report results of numerical experiments. In Section 5, we make some concluding remarks.

2. Sum of one fraction and a convex function

Throughout this section, we assume that $p = 1$. In this case, problem (1) reduces to the form

$$\text{minimize} \quad h(x) + \frac{f(x)}{g(x)} \qquad \text{subject to} \quad x \in \mathcal{S}. \tag{3}$$

Here, $f, g, h : \mathcal{S} \mapsto \mathbb{R}$ are functions that satisfy the conditions specified in Assumption 1, i.e., f and h are convex, g is concave, and $f(x) \geqslant 0$ and $g(x) > 0$ for $x \in \mathcal{S}$. For any fixed $r > 0$, let

$$x(r) := \arg\min \left\{ h(x) + \frac{f(x)}{r} \;\middle|\; g(x) \geqslant r \text{ and } x \in \mathcal{S} \right\} \tag{4}$$

and

$$q(r) := h(x(r)) + \frac{f(x(r))}{r}. \tag{5}$$

For $r > \max\{ g(x) \mid x \in \mathcal{S} \}$, the feasible set in (4) is empty, and in this case, we set $q(r) := \infty$. Note that $x(r)$ is not necessarily unique, but, of course, $q(r)$ is. From the definition of q, it is obvious that $x(r^*)$ solves (3) if, and only if, r^* minimizes q. Thus, problem (3) is reduced to the one-dimensional problem of minimizing the function q.

Determining $x(r)$ for a given value $r > 0$ is a convex optimization problem, which can be solved by several methods.

If a separation oracle for $\mathcal{S}$ ($\subset \mathbb{R}^n$) is given, the evaluation of q for a given value of r can be done (up to a given precision) by the ellipsoid method. Here, by ‘separation oracle’, we mean a subprogram that accepts as input any vector $x \in \mathbb{R}^n$ and produces as output either the information ‘$x \in \mathcal{S}$’, or a vector $h \in \mathbb{R}^n$, $h \neq 0$, with $h^T y \leqslant h^T x$ for all $y \in \mathcal{S}$. In the second case, the vector h defines a hyperplane that ‘separates’ x from $\mathcal{S}$.

If self-concordant barrier functions for the sets

$$\left\{ x \in \mathcal{S} \;\middle|\; h(x) + \frac{f(x)}{r} \leqslant \lambda \text{ and } g(x) \geqslant r \right\}$$

for real numbers λ are known, then $q(r)$ can also be evaluated by an interior-point method. Here, a barrier function for a convex set $\mathcal{C}$ is a function that is convex and finite in the interior of $\mathcal{C}$, and goes to infinity as x approaches the boundary of $\mathcal{C}$. The notion of self-concordance was first introduced in Nesterov and Nemirovskii (1994). Roughly speaking, self-concordance is defined as a local Lipschitz condition of the Hessian of the barrier function. As shown in Nesterov and Nemirovskii (1994), many convex sets possess easily computable self-concordant barrier functions, and the concept of interior-point methods based on self-concordance is a very general approach.

We remark that for the special case of a constant function h and $p = 2$ ratios, problem (1) can be reduced to a problem of the form (3), i.e., with only one ratio, by means of the Charnes–Cooper transformation (Charnes and Cooper, 1962); (see, e.g., Cambini et al., 1989). A self-concordant barrier function for the conic hull introduced by this transformation is discussed in Freund et al. (1996). In general, when $p > 1$ and h is constant, the Charnes–Cooper transformation can be used to reduce problem (1) to a sum-of-ratios problem with $p-1$ ratios. This simple reduction may be crucial for algorithms whose computational costs grow rapidly with the number of ratios. For example, given a sum-of-ratios problem with $p = 2$ and $h = 0$, it will be more efficient to first employ the Charnes–Cooper transformation and then apply the algorithm of the present paper to the reformulation with $p = 1$, rather than using the same algorithm for the solution of the original problem with $p = 2$.

2.1. PROPERTIES OF THE FUNCTION q

Next, we recall some well-known properties of the function q given by (4) and (5).

Let $0 < r < s < t$ be given. Set

$$\sigma := \frac{s-r}{t-r} \in (0, 1) \quad \text{and} \quad \overline{x} := (1-\sigma)x(r) + \sigma x(t).$$

Then, $s = (1-\sigma)r + \sigma t$, and by the convexity of f, we have

$$\frac{f(\overline{x})}{s} \leqslant \frac{(1-\sigma)f(x(r)) + \sigma f(x(t))}{(1-\sigma)r + \sigma t} \leqslant \max\left\{ \frac{f(x(r))}{r}, \ \frac{f(x(t))}{t} \right\}. \tag{6}$$

Similarly, the convexity of h implies that

$$h(\overline{x}) \leqslant (1-\sigma)h(x(r)) + \sigma h(x(t)) \leqslant \max\{ h(x(r)), \ h(x(t)) \}. \tag{7}$$

By the concavity of g, it also follows that $g(\overline{x}) \geqslant s$. Hence, $\overline{x}$ is feasible for (4), and

$$q(s) \leqslant h(\overline{x}) + \frac{f(\overline{x})}{s}. \tag{8}$$

In spite of (6), (7), and (8), the function q is not quasi-convex, i.e., in general it may happen that

$$q(s) \not\leqslant \max\{ q(r), q(t) \}.$$

Note that if q were quasi-convex, problem (3) could be solved in polynomial time by using a golden-mean search for q.

In view of the above derivation, we may still ask ourselves whether the function q may 'smooth out' some of the local minimizers of (3), and whether minimizing q might be easier than solving problem (3) directly (assuming that we can evaluate

q and its derivatives). The observation that the function q is not necessarily simpler than (3) is illustrated in Figure 1 below, which depicts the function q for a special case where $\mathcal{S}$ is just a real interval. This plot shows that q may exhibit a very 'irregular' behavior.

2.2. $\mathcal{NP}$-COMPLETENESS

Next, we prove that problem (3) is 'essentially' $\mathcal{NP}$-complete. To this end, we show that a well-known $\mathcal{NP}$-complete problem, namely the following knapsack problem, can be recast as a special instance of problem (3).

Knapsack problem:
Let an integer $d > 1$, weights $w_1, w_2, \ldots, w_d > 0$, a weight limit $\overline{w} > 0$, and costs $c_1, c_2, \ldots, c_d > 0$ be given. Set $I := \{1, 2, \ldots, d\}$. The problem is to find a subset $I' \subset I$ such that $\sum_{i \in I'} c_i$ is maximized subject to the constraint $\sum_{i \in I'} w_i \leqslant \overline{w}$.

For a discussion of the knapsack problem and a proof of its $\mathcal{NP}$-completeness, we refer the reader to Garey and Johnson (1979).

Our result on the $\mathcal{NP}$-completeness of problem (3) can now be stated as follows.

THEOREM 2. *Problem* (3) *is $\mathcal{NP}$-complete in the following sense. Let the data of a knapsack problem with $d \in \mathbb{N}$ weights be given. There exists a convex, piecewise linear function f, a linear function g, and a linear function h defined on the interval $\mathcal{S} = [1, 2^d]$ such that f, g, h, and their respective derivatives can be evaluated in polynomial time, f, g, and h take values of polynomial size, and solving problem* (3) *is equivalent to solving the given knapsack problem.*

REMARK 3. The right endpoint 2^d of the interval $\mathcal{S}$ in Theorem 2 is not polynomial. At first sight, this might lead to the impression that the reduction of a knapsack problem to problem (3) is exponential. This, of course, is not the case. Indeed, just as in the case of linear programs, which may also involve non-polynomial upper or lower bounds, one only needs polynomiality in the coding length of the problem. The coding length of problem (3) is at least d, and hence the coding length of the endpoint 2^d is in fact polynomial in the coding length of the problem. Finally, note that if the function f/g were convex, then an ϵ-approximation to problem (3) could be computed in polynomial time. The $\mathcal{NP}$-completeness in Theorem 2 does not result from the size of the endpoint 2^d of $\mathcal{S}$, but from the lack of convexity.

Proof of Theorem 2. Let $d \in \mathbb{N}$, weights $w_1, w_2, \ldots, w_d$, a weight limit $\overline{w} > 0$, and costs $c_1, c_2, \ldots, c_d > 0$ be the given data of a knapsack problem. From this data, we now construct a special problem of the form (3) that is 'equivalent' to the knapsack problem.

To this end, we first enumerate the 2^d subsets of $I = \{1, 2, \dots, d\}$, by simply counting from 1 to 2^d in the binary system. Then, for each subset I' of I, there exists an index $1 \leqslant k \leqslant 2^d$ such that $I' = I_k$ is the k-th subset of the enumeration. We can determine I_k just by knowing its index k, and without looking at any other subset. We can also determine the weight $\sum_{i \in I_k} w_i$ of the k-th subset just by knowing the index k. For $1 \leqslant k \leqslant 2^d$, we set

$$\eta_k := \begin{cases} 1 & \text{if } \displaystyle\sum_{i \in I_k} w_i > \overline{w}, \\ 1 - \displaystyle\sum_{i \in I_k} c_i/\overline{c} & \text{otherwise,} \end{cases} \qquad \text{and} \qquad \epsilon_k := \frac{k\eta_k}{2^d},$$

where $\overline{c} := \sum_{i=1}^{d} c_i$. Solving the knapsack problem is then equivalent to finding

$$\min_{1 \leqslant k \leqslant 2^d} \eta_k. \tag{9}$$

For later use, we note that

$$0 \leqslant \epsilon_k \leqslant 1 \quad \text{for all} \quad 1 \leqslant k \leqslant 2^d. \tag{10}$$

Next, we set $\mathcal{S} := [1, 2^d]$ and define functions $f, g, h : \mathcal{S} \mapsto \mathbb{R}$ as follows. The functions g and h are the linear functions given by

$$g(x) \equiv x \quad \text{and} \quad h(x) \equiv -x \quad \text{for all} \quad x \in \mathcal{S}. \tag{11}$$

The function f is defined as the piecewise linear interpolant through the points $(k, k^2 + \epsilon_k)$, $1 \leqslant k \leqslant 2^d$. Hence, on each interval $k \leqslant x \leqslant k+1$, where $1 \leqslant k < 2^d$, f is given by

$$f(x) \equiv \left(k^2 + \epsilon_k\right)(k + 1 - x) + \left((k+1)^2 + \epsilon_{k+1}\right)(x - k). \tag{12}$$

Using (10), one readily verifies that the function f is convex on $\mathcal{S}$. Clearly, given any $x \in \mathcal{S}$, it is possible to evaluate $f(x)$ in $\mathcal{O}(d)$ arithmetic operations, and the number of digits needed to represent the function values $k^2 + \epsilon_k$ are at most $2d$ plus the number of digits needed to evaluate $\sum_{i \in I_k} c_i/\overline{c}$.

Finally, we show that for the set $\mathcal{S}$ and the functions f, g, and h just defined, the minimizer of (3) is the index k of a k-th subset $I_k \subset I$ that solves the knapsack problem. Let $1 \leqslant k < 2^d$ and consider the objective function of (3) for $x \in [k, k+1]$. By (10)–(12), the second derivative of the objective function satisfies

$$\begin{aligned} \frac{d^2}{dx^2}\left(h(x) + \frac{f(x)}{g(x)}\right) &= \frac{2}{x^3}\left(-k(k+1) + \epsilon_k(k+1) - \epsilon_{k+1}k\right) \\ &\leqslant \frac{2}{x^3}\left(-k(k+1) + k + 1\right) = \frac{2}{x^3}\left(1 - k^2\right) \leqslant 0 \end{aligned}$$

for all $x \in [k, k+1]$. This shows that the objective function of (3) is concave on $[k, k+1]$, and thus its minimum over $[k, k+1]$ is attained at $x = k$ or $x = k+1$. By (11) and (12), the corresponding function values are

$$\frac{\epsilon_k}{k} = \frac{\eta_k}{2^d} \quad \text{for} \quad x = k, \quad \text{and} \quad \frac{\epsilon_{k+1}}{k+1} = \frac{\eta_{k+1}}{2^d} \quad \text{for} \quad x = k+1.$$

Therefore, problem (3) is equivalent to (9), which in turn is equivalent to solving the knapsack problem. □

For the special instance of problem (3) constructed in the proof of Theorem 2, the evaluation of the associated function (5), $q(r)$, is particularly simple. Indeed, let $x \geqslant r$ and $x \in [k, k+1]$ for some $1 \leqslant k < 2^d$. Then, by (10)–(12),

$$\begin{aligned}\frac{d}{dx}\left(h(x) + \frac{f(x)}{r}\right) &= -1 + \frac{2k + 1 + \epsilon_{k+1} - \epsilon_k}{r} \\ &\geqslant -1 + \frac{2k}{r} \geqslant -1 + \frac{k+1}{r} \geqslant -1 + \frac{x}{r} \geqslant 0.\end{aligned}$$

This shows that the objective function in (4) is monotonically increasing for $x = g(x) \geqslant r$ and $x \in \mathcal{S}$. Therefore, the minimum in (4) is attained for $x(r) = r$, and the function $q(r)$ in (5) is identical to the objective function of (3). In Figure 1, we plot the function q for the case of the knapsack problem with $d = 4$, random values $w_i, c_i \in (0, 1)$, $1 \leqslant i \leqslant d$, and weight limit $\overline{w} = 0.8 \sum_{i=1}^{d} w_i$.

Figure 1 displays an example where minimizing q is identical to solving problem (3). In general, however, we may anticipate that the structure of the higher-dimensional problem (3) is far more complicated than the scalar function q. We propose an approach for solving problem (3) by evaluating q for various values of r and exploiting Lipschitz properties of q.

We emphasize that in the case where the function q has very many local minimizers of approximately the same magnitude (as in the class of problems constructed in the proof of Theorem 2), any approach for solving problem (1) will necessarily be very slow (unless $\mathcal{P} = \mathcal{NP}$).

2.3. A GLOBAL MINIMIZATION METHOD

If f, g, and h are smooth, due to the structure of $\mathcal{S}$, the function q is generally a piecewise smooth function. To compute a global minimizer r^* of q, we construct a lower-bound function $\underline{q}(r) \leqslant q(r)$ and then minimize $\underline{q}$.

The function $\underline{q}$ depends on a partition, $0 < r^{(1)} < r^{(2)} < \cdots < r^{(k)}$, where we assume that $r^{(1)} \leqslant r^*$ and $r^{(k)} \geqslant r^*$. Note that

$$0 < \min_{x \in \mathcal{S}} g(x) \;\leqslant\; r^* \;\leqslant\; \max_{x \in \mathcal{S}} g(x), \tag{13}$$

so that a value for $r^{(1)}$ can be obtained from a given lower bound of g on $\mathcal{S}$, and a value for $r^{(k)}$ by solving the concave maximization problem in (13).

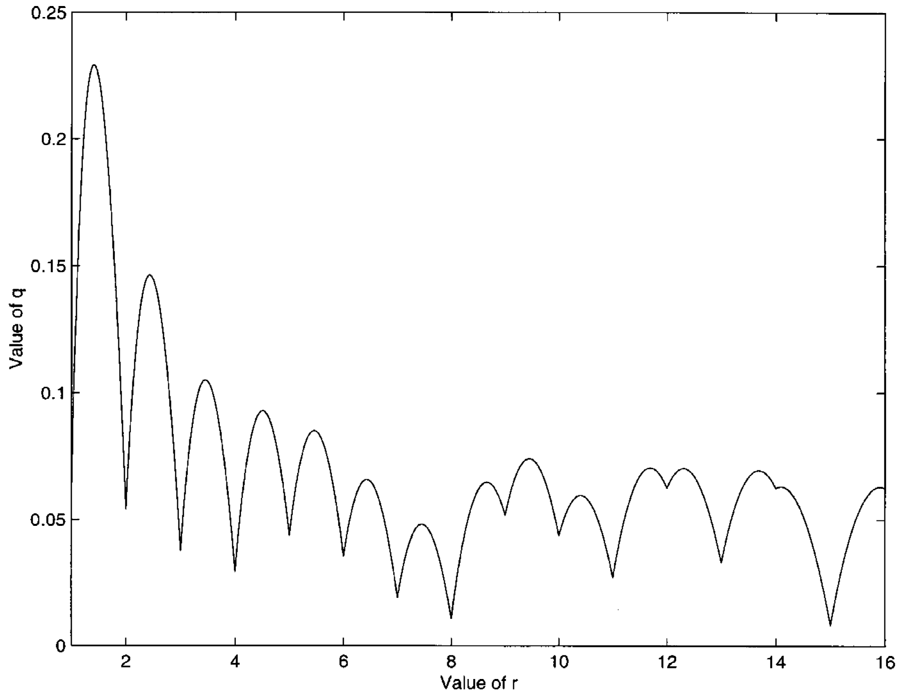

Figure 1. The objective function with random w_i's and c_i's.

Let some i and $0 < r^{(i)} < r^{(i+1)}$ be given. Define a lower-bound number

$$\underline{q_i} \leqslant \min \left\{ h(x) + \frac{f(x)}{r^{(i+1)}} \;\middle|\; x \in \mathcal{S} \text{ and } g(x) \geqslant r^{(i)} \right\} \tag{14}$$

so that $\underline{q_i} \leqslant q(r)$ for $r \in [r^{(i)},\ r^{(i+1)}]$. Note that evaluating the right-hand side of (14) amounts to solving a convex optimization problem. Let $\underline{\mathrm{x}}_i$ be a solution of the minimization problem in (14). It follows that

$$h(x(r)) + \frac{f(x(r))}{r^{(i+1)}} \geqslant h(\underline{\mathrm{x}}_i) + \frac{f(\underline{\mathrm{x}}_i)}{r^{(i+1)}} \;\geqslant\; \underline{q_i},$$

$$h(x(r)) + \frac{f(x(r))}{r^{(i)}} \geqslant h(x(r^{(i)})) + \frac{f(x(r^{(i)}))}{r^{(i)}} \;=\; q(r^{(i)}).$$

Using these two inequalities, for all $r \in [r^{(i)}, r^{(i+1)}]$, we get

$$\begin{aligned}
q(r) =& h(x(r)) + \frac{f(x(r))}{r} \\
=& \frac{r^{(i+1)}}{r}\ \frac{r - r^{(i)}}{r^{(i+1)} - r^{(i)}} \left(h(x(r)) + \frac{f(x(r))}{r^{(i+1)}} \right) \\
&+ \frac{r^{(i)}}{r}\ \frac{r^{(i+1)} - r}{r^{(i+1)} - r^{(i)}} \left(h(x(r)) + \frac{f(x(r))}{r^{(i)}} \right) \\
\geqslant& \frac{r^{(i+1)}}{r}\ \frac{r - r^{(i)}}{r^{(i+1)} - r^{(i)}}\ \underline{q_i} + \frac{r^{(i)}}{r}\ \frac{r^{(i+1)} - r}{r^{(i+1)} - r^{(i)}}\ q(r^{(i)}) \\
=& q(r^{(i)}) + \frac{r^{(i+1)}}{r^{(i)}}\ \frac{r - r^{(i)}}{r^{(i+1)} - r^{(i)}} \left(\frac{r^{(i)}}{r} \left(\underline{q_i} - q(r^{(i)}) \right) \right) \\
\geqslant& q(r^{(i)}) + \frac{r^{(i+1)}}{r^{(i)}}\ \frac{r - r^{(i)}}{r^{(i+1)} - r^{(i)}} \left(\underline{q_i} - q(r^{(i)}) \right) \qquad (15) \\
=& q(r^{(i)}) + \underline{q_i}'(r - r^{(i)}), \qquad (16)
\end{aligned}$$

where

$$\underline{q_i}' := \frac{r^{(i+1)}}{r^{(i)}}\ \frac{\underline{q_i} - q(r^{(i)})}{r^{(i+1)} - r^{(i)}}.$$

Note that the inequality (15) follows from $r^{(i)}/r \leqslant 1$ and $\underline{q_i} - q(r^{(i)}) \leqslant 0$.

The bound (16) proves left-sided Lipschitz continuity of q. Indeed, near $r = r^{(i)}$, the above bound is close to the value $q(r^{(i)})$. However, for $r = r^{(i+1)}$, the bound reduces to $\underline{q_i}$, which is lower than the value $q(r^{(i+1)})$.

Note that a bound of the form (15) with $q(r^{(i+1)})$ in place of $q(r^{(i)})$ is not possible. It may occur that $q(r^{(i+1)}) \gg q(r^{(i)})$. Intuitively, this will happen when $\mathcal{S} \cap \{ x \mid g(x) \geqslant r^{(i+1)} \}$ no longer contains points for which f or h are reasonably small but $\mathcal{S} \cap \{ x \mid g(x) \geqslant r^{(i)} \}$ does. In this case, the Lipschitz constant for q from the right may be much larger than the one from the left. To determine a suitable Lipschitz property from the right, we define the function

$$\tilde{q}_i(r) := \min \left\{ h(x) + \frac{f(x)}{r^{(i+1)}} \;\middle|\; x \in \mathcal{S} \text{ and } g(x) \geqslant r \right\}. \qquad (17)$$

Observe that $\tilde{q}_i$ is convex (in r). Moreover, $\tilde{q}_i$ satisfies $\tilde{q}_i(r) \leqslant q(r)$ for $r \leqslant r^{(i+1)}$, and $\tilde{q}_i(r^{(i+1)}) = q(r^{(i+1)})$. We remark that when evaluating $q(r^{(i+1)})$, the problem (17) with $r = r^{(i+1)}$ is solved, and the Lagrange multiplier – denoted by λ_g in the sequel – corresponding to the constraint $g(x) \geqslant r^{(i+1)}$ can also be computed. Indeed, interior-point methods can be implemented so that such a multiplier is obtained at no extra cost. The Lagrange multiplier leads to the bound

$$q(r) \geqslant \tilde{q}_i(r) \geqslant q(r^{(i+1)}) + \lambda_g(r - r^{(i+1)}) \qquad (18)$$

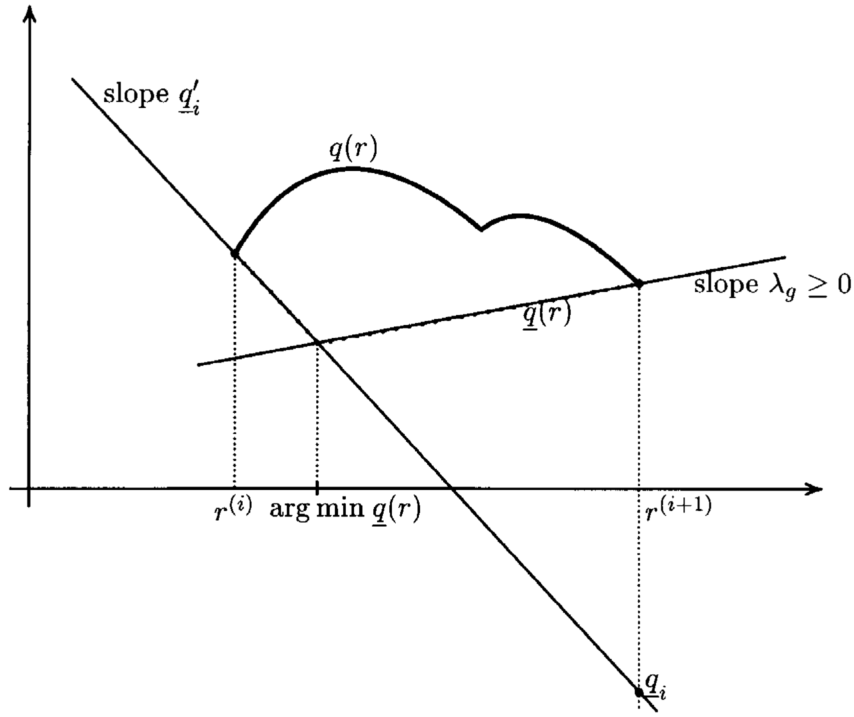

Figure 2. The functions q and $\underline{q}$.

for $r \leqslant r^{(i+1)}$; see, e.g., Theorem VII.3.3.2 in Hiriart-Urruty and Lemarechal (1993). The lower-bound function $\underline{q}(r)$ is then defined for $r \in [r^{(i)}, r^{(i+1)}]$ as the maximum of the bounds (16) and (18),

$$\underline{q}(r) := \max\left\{ q(r^{(i)}) + \underline{q_i}'(r - r^{(i)}),\ q(r^{(i+1)}) + \lambda_g(r - r^{(i+1)}) \right\}.$$

A simple method for solving problem (3) then proceeds as follows. Given k points

$$0 < r^{(1)} < r^{(2)} < \cdots < r^{(k)}, \quad \text{with} \quad r^{(1)} \leqslant r^* \leqslant r^{(k)}, \tag{19}$$

a new point $\hat{r}$ from the interval $(r^{(i)}, r^{(i+1)})$ that contains a minimizer of $\min\{\underline{q}(r) \mid r \in [r^{(1)}, r^{(k)}]\}$ is chosen. Then, $\hat{r}$ is inserted into the list (19) (thus k is increased by one), and the process is repeated.

Note that the update of $\underline{q}(r)$ only involves the interval between $r^{(i)}$ and $r^{(i+1)}$ neighboring $\hat{r}$. This interval is split into two subintervals $[r^{(i)}, \hat{r}]$ and $[\hat{r}, r^{(i+1)}]$, and the minimum of $\underline{q}(r)$ is evaluated over both intervals. In particular, the effort for minimizing $\underline{q}$ merely consists of bookkeeping. Figure 2 gives an example of the bounds leading to $\underline{q}(r)$.

Note that the slopes of $q(r)$ and $\underline{q}(r)$ may be of opposite sign, so that in the interior of $[r^{(i)}, r^{(i+1)}]$ the function $\underline{q}(r)$ may not be a good approximation to $q(r)$.

Hence, $\hat{r} := \arg\min\{\underline{q}(r) \mid r \in [r^{(1)}, r^{(k)}]\}$ may be a poor choice. A more reliable choice used in our numerical examples below is $\hat{r} := \frac{1}{2}(r^{(i)} + r^{(i+1)})$.

To keep the evaluation of q at moderate costs, it suffices to compute only approximations to $q(\hat{r})$ and $\tilde{q}'(\hat{r})$, along with some error estimates. Interior-point methods are particularly suitable for computing an approximate solution $\hat{x}(r)$ of the convex problem (4), along with a certified error bound of the form $|q(r) - h(\hat{x}(r)) - f(\hat{x}(r))/r| \leqslant \epsilon$. The computation of $\hat{x}(r)$ takes at most $\mathcal{O}(\log(1/\epsilon))$ iterations provided that a self-concordant barrier function for $\mathcal{S}$ and for the level sets of the functions f, g, and h is known. This observation is the key point for our proposed algorithm. Next, we present a statement of the algorithm.

ALGORITHM 4. (Conceptual overall algorithm for $p = 1$.)

INPUT. *Functions f, g, h and a compact convex set $\mathcal{S}$ defining the single-ratio problem* (3).
A stopping tolerance $\epsilon > 0$.

Step 0. *Determine $r^{(1)}$ and $r^{(2)}$ with*

$$0 < r^{(1)} \leqslant \min_{x \in \mathcal{S}} g(x) \quad \textit{and} \quad r^{(2)} \geqslant \max_{x \in \mathcal{S}} g(x).$$

If no such value $r^{(1)}$ exists: STOP, the problem violates Assumption 1.
Otherwise, compute $q(r^{(1)})$, $q(r^{(2)})$, and the Lagrange multipliers $\lambda_g^{(\cdot)}$ for g.
Set $k = 2$ (number of 'support points' $r^{(\cdot)}$).
Set $i = 1$ (interval $(r^{(i)}, r^{(i+1)})$ containing $\arg\min \underline{q}(r)$).

Step 1. *Set $\hat{r} = \frac{1}{2}(r^{(i)} + r^{(i+1)})$.*

Step 2. *Compute $q(\hat{r})$ along with the Lagrange multiplier $\hat{\lambda}_g$ for g.*

Step 3. *Based on* (18), *evaluate*

$$\arg \min_{r \in [r^{(i)}, \hat{r}]} \underline{q}(r) \quad \textit{and} \quad \arg \min_{r \in [\hat{r}, r^{(i+1)}]} \underline{q}(r).$$

Step 4. *Increase k by one, and insert $\hat{r}$ into the list of $r^{(\cdot)}$'s.*

Step 5. *Find $i \leqslant k - 1$ such that $\tilde{r} := \arg\min_r \underline{q}(r) \in (r^{(i)}, r^{(i+1)})$.*

Step 6. *If $\underline{q}(\tilde{r}) \geqslant \min_{1 \leqslant \iota \leqslant k} q(r^{(\iota)}) - \epsilon$, then STOP:*
$\tilde{r}$ is an approximate minimizer.
Otherwise, return to Step 1.

We remark that, in Step 3 of Algorithm 4, the bound $\underline{q}(r)$ may either be obtained by setting $\underline{q}_i = -\infty$, or by solving an additional problem of the form (14). The latter case is more expensive and results in a better bound for $\underline{q}(r)$ since both (16) and (18) are used. In most cases (unless λ_g in (18) is 'overly large'), it is more efficient to rely on (18) only, and not to solve (14).

Note that the minimizers of the lower-bound function $\underline{q}(r)$ computed in Step 3 of Algorithm 4 can be stored in a heap, so that Step 5 merely consists of selecting the first element from this heap.

Finally, we remark that, in practice, the feasible set $\mathcal{S}$ will usually be of the form

$$\mathcal{S} = \left\{ x \in \mathbb{R}^n \;\middle|\; b_i(x) \leqslant 0 \text{ for all } i = 1, 2, \ldots, m \right\},$$

where $b_1, b_2, \ldots, b_m$ are given convex functions.

3. Minimizing the sum of several fractions

In this section, we return to the general problem (1) of minimizing the sum of a convex function and p ratios. The basic idea for solving problem (1) is similar to the special case $p = 1$ treated in Section 2.

Let $p \geqslant 2$, and again let Assumption 1 be satisfied. In this case, $r = [r_1 \quad r_2 \quad \cdots r_p]^T \in \mathbb{R}^p$ is a vector of p parameters. In analogy to the definitions (4) and (5) of $x(r)$ and $q(r)$ in the case $p = 1$, we set

$$\begin{aligned} x(r) &:= \arg\min \left\{ h(x) + \sum_{j=1}^{p} \frac{f_j(x)}{r_j} \;\middle|\; x \in \mathcal{S}(r) \right\}, \\ &\text{where } \mathcal{S}(r) := \left\{ x \in \mathcal{S} \;\middle|\; g_j(x) \geqslant r_j \text{ for all } j = 1, 2, \ldots, p \right\}, \end{aligned} \tag{20}$$

and

$$q(r) := h(x(r)) + \sum_{j=1}^{p} \frac{f_j(x(r))}{r_j}.$$

Initially, we assume that vectors $r^{(1)}$ and $r^{(2)}$ are computed such that there is a minimizer r^* of q satisfying $r^{(1)} \leqslant r^* \leqslant r^{(2)}$. (As usual, the $\leqslant$-sign is understood component wise.) Each component of $r^{(1)}$ and $r^{(2)}$ can be computed separately as in the case $p = 1$.

Now let r, $r^{(i)}$, $r^{(i+1)}$, and some direction $\Delta r \in \mathbb{R}^p$ be given such that the relations

$$r^{(i)} \leqslant r \leqslant r^{(i+1)} \quad \text{and} \quad r^{(i)} \leqslant r + \Delta r \leqslant r^{(i+1)}$$

are satisfied. The bounds (16) and (18) can be generalized to provide bounds for $q(r + \Delta r)$. We split $\Delta r = \Delta r^+ - \Delta r^-$ with $\Delta r^+,\ \Delta r^- \geqslant 0$ and $(\Delta r^+)^T \Delta r^- = 0$. Let $\lambda_j \geqslant 0$ be the Lagrange multipliers for the constraints $g_j(x) \geqslant r_j$ in (20). By

Theorem VII.3.3.2 in Hiriart-Urruty and Lemarechal (1993), a lower bound for q is given by

$$q(r - \Delta r^-) \geqslant q(r) - \lambda^T \Delta r^-. \tag{21}$$

To obtain a lower bound for q in direction Δr^+, we define the value

$$\underline{q_{i,i+1}} \leqslant \min \left\{ h(x) + \sum_{j=1}^{p} \frac{f_j(x)}{r_j^{(i+1)}} \;\middle|\; x \in \mathcal{S}\left(r^{(i)}\right) \right\}.$$

It then follows that

$$\begin{aligned} q(r) =& h(x(r)) + \sum_{j=1}^{p} \frac{f_j(x(r))}{r_j} \\ =& h(x(r)) + \sum_{j=1}^{p} \underbrace{\frac{r_j^{(i)}}{r_j} \frac{r_j^{(i+1)} - r_j}{r_j^{(i+1)} - r_j^{(i)}}}_{=1-\nu_j} \frac{f_j(x(r))}{r_j^{(i)}} \\ &+ \sum_{j=1}^{p} \underbrace{\frac{r_j^{(i+1)}}{r_j} \frac{r_j - r_j^{(i)}}{r_j^{(i+1)} - r_j^{(i)}}}_{=:\nu_j} \frac{f_j(x(r))}{r_j^{(i+1)}} \\ =& (1-\overline{\nu}) \underbrace{\left(h(x(r)) + \sum_{j=1}^{p} \frac{f_j(x(r))}{r_j^{(i)}} \right)}_{\geqslant q(r^{(i)})} \\ &+ \overline{\nu}\, h(x(r)) + \sum_{j=1}^{p} \left(\underbrace{(\overline{\nu} - \nu_j)}_{\geqslant 0} \underbrace{\frac{f_j(x(r))}{r_j^{(i)}}}_{\geqslant f_j(x(r))\,/\,r_j^{(i+1)}} + \nu_j \frac{f_j(x(r))}{r_j^{(i+1)}} \right), \end{aligned}$$

where

$$\overline{\nu} := \max_{1 \leqslant j \leqslant p} \nu_j = \max_{1 \leqslant j \leqslant p} \frac{r_j^{(i+1)}}{r_j} \frac{r_j - r_j^{(i)}}{r_j^{(i+1)} - r_j^{(i)}} \leqslant \max_{1 \leqslant j \leqslant p} \frac{r_j^{(i+1)}}{r_j^{(i)}} \frac{r_j - r_j^{(i)}}{r_j^{(i+1)} - r_j^{(i)}}.$$

Combining the above relations, we get

$$\begin{aligned} q(r) &\geqslant (1-\overline{\nu})\, q(r^{(i)}) + \overline{\nu} \left(h(x(r)) + \sum_{j=1}^{p} \frac{f_j(x(r))}{r_j^{(i+1)}} \right) \\ &\geqslant (1-\overline{\nu})\, q(r^{(i)}) + \overline{\nu}\, \underline{q_{i,i+1}}. \end{aligned} \tag{22}$$

This bound is analogous to the one for $p = 1$ given in (16).

In (22), we may replace $r^{(i)}$ by $r - \Delta r^-$, $q(r^{(i)})$ by $q(r) - \lambda^T \Delta r^-$, and r by $(r - \Delta r^-) + \Delta r^+$ to obtain the new bound

$$q(r + \Delta r) \geqslant (1 - \overline{\nu})(q(r) - \lambda^T \Delta r^-) + \overline{\nu}\underline{q_{i,i+1}}$$

with

$$\overline{\nu} := \max_{1 \leqslant j \leqslant p} \frac{r_j^{(i+1)}}{r_j + \Delta r_j^+} \frac{\Delta r_j^+}{r_j^{(i+1)} - r_j} \leqslant \max_{1 \leqslant j \leqslant p} \frac{r_j^{(i+1)}}{r_j} \frac{\Delta r_j^+}{r_j^{(i+1)} - r_j}.$$

Based on this bound, we can define an anisotropic trust region about each point r, as long as some lower and some upper limits (like $r^{(i)}$ and $r^{(i+1)}$ in the previous derivation) are given. The union of the trust regions about all support points $r^{(i)}$ forms a Voronoi diagram in $\mathbb{R}^p$, the vertices of which contain the candidates for the minimizer of the lower-bound function $\underline{q}(r)$. For a definition of Voronoi diagrams, their properties, and algorithms for their numerical computation, we refer the reader to Aurenhammer (1991); Fortune (1997). As in the case $p = 1$, these candidates for the minimizer of $\underline{q}(r)$ may not result in the best choice for inserting a new value $\hat{r}$ somewhere between the known points $r^{(i)}$. In addition, the computation of the vertices of the Voronoi diagram is complicated and expensive. We propose a simpler scheme based on bounds analogous to (18) where the lower-bound function $\underline{q}(r)$ is defined in the box

$$\mathcal{B}^{(i)} := \left\{ r \in \mathbb{R}^p \;\middle|\; r_j \in [r_j^{(i)-}, r_j^{(i)}] \text{ for } 1 \leqslant j \leqslant p \right\}$$

with given vectors $r_j^{(i)-} < r_j^{(i)}$. For $r \in \mathcal{B}^{(i)}$, we obtain from (21) that $\underline{q}(r) := q(r^{(i)}) + \lambda^T(r - r^{(i)}) \leqslant q(r)$.

Next, we summarize the resulting overall algorithm.

ALGORITHM 5. (Conceptual overall algorithm for $p \geqslant 2$.)

INPUT. *Functions* $f_1, f_2, \ldots, f_p, g_1, g_2, \ldots, g_p, h$, *and a compact convex set* $\mathcal{S}$ *defining the multi-ratio problem* (1).
A stopping tolerance $\epsilon > 0$.

Step 0. *For all* $1 \leqslant j \leqslant p$, *determine*

$$0 < r_j^{(1)} \leqslant \min_{x \in \mathcal{S}} g_j(x) \quad \textit{and} \quad r_j^{(2)} \geqslant \max_{x \in \mathcal{S}} g_j(x).$$

If $r_j^{(1)}$ *does not exist for some* $1 \leqslant j \leqslant p$*: STOP, the problem violates Assumption* 1.

Otherwise, compute $q(r^{(1)})$, $q(r^{(2)})$, *and the Lagrange multipliers* $\lambda_j^{(\cdot)}$ *for* *the* g_j*'s.*

Set $k = 2$ (*number of 'support points'* $r^{(\cdot)}$).

Set $i = 2$ *and* $r^{(2)^-} := r^{(1)}$ (*the box* $r^{(i)^-} \leqslant r \leqslant r^{(i)}$ *containing* $\arg\min \underline{q}(r)$).

Step 1. *Set* $l = \arg \max_{1\leqslant j\leqslant p} \lambda_j^{(i)}(r_j^{(i)} - r_j^{(i)^-})$ (*the index where a split pays most*) *and define* $\hat{r}$ *by*

$$\hat{r}_j = \begin{cases} r_j^{(i)} & \text{for } j \neq l, \\ \frac{1}{2}(r_j^{(i)} + r_j^{(i)-}) & \text{for } j = l. \end{cases}$$

Step 2. *Compute* $q(\hat{r})$ *along with the Lagrange multiplier* $\hat{\lambda}_j$ *for each function* g_j.

Step 3. *Based on* (21), *evaluate*

$$\arg \min_{r\in\mathcal{B}^{(i)}:\ r_l\in[r_l^{(i)^-},\hat{r}_l]} \underline{q}(r) \quad \text{and} \quad \arg \min_{r\in\mathcal{B}^{(i)}:\ r_l\in[\hat{r}_l,r_l^{(i)}]} \underline{q}(r).$$

Step 4. *Increase* k *by one, insert* $\hat{r}$ *into the list of* $r^{(\cdot)}$*'s splitting* $\mathcal{B}^{(i)}$ *along the hyperplane* $r_l = \hat{r}_l$ *into two boxes* (*one for* $\hat{r}$ *and one for* $r^{(i)}$).

Step 5. *Find* $i \leqslant k$ *such that* $\tilde{r} := \arg\min_r \underline{q}(r) \in \mathcal{B}^{(i)}$.

Step 6. *If* $\underline{q}(\tilde{r}) \geqslant \min_{1\leqslant\iota\leqslant k} q(r^{(\iota)}) - \epsilon$, *then STOP:*
$\tilde{r}$ *is an approximate minimizer.*

Otherwise, return to Step 1.

4. Numerical experiments

Algorithm 4 for minimizing the sum of a convex function and of $p = 1$ convex-concave fraction has been implemented in Matlab. For the solution of the convex subproblems, we use the interior-point method described in (Jarre and Saunders, 1995). As we have seen in Figure 1, the resulting problem (3) may be very complicated, and may have very many local minimizers. Nevertheless, we anticipate that the parameterization with respect to r will smooth out many of the local minimizers of (3) and thus result in a function q that is easier to minimize than the objective function of the original problem (3).

In this section, we report numerical results of Algorithm 4 applied to certain examples with random data. In this case, the expectation that the function q is easier to minimize than the original problem (3) was fully met. In fact, the function q appeared to be unimodal with respect to r for these examples.

Our test examples are minimization problems of the form (3), where

$$\begin{aligned} h(x) &= \tfrac{1}{2}x^T Hx + h^T x + \kappa, \quad f(x) = \tfrac{1}{2}x^T Fx + f^T x + \varphi, \\ g(x) &= -\tfrac{1}{2}x^T Gx - g^T x - \gamma, \\ \mathcal{S} &= \left\{ x \in \mathbb{R}^n \;\middle|\; x^T D_i x + d_i^T x + \delta_i \leqslant 0,\ i = 1, 2, \ldots, m \right\}. \end{aligned} \tag{23}$$

Here, the matrices H, F, G and $D_1, D_2, \dots, D_m$ are constructed to be positive semidefinite. Therefore, in (23), the functions h and f are convex, the function g is concave, and the feasible set $\mathcal{S}$ is convex. The data for (23) is chosen randomly as follows. For each matrix D_i, we first generated a random lower bidiagonal matrix L_i the nonzero entries of which are uniformly distributed in $[-1, 1]$, and then we computed $D_i := L_i L_i^T$. This guarantees that each D_i is a positive semidefinite tridiagonal matrix. Similarly, H, F, and G are constructed as random positive semidefinite tridiagonal matrices. In (23), h, f, g, and $d_1, d_2, \dots, d_m$ are vectors that were also generated randomly. Furthermore, the scalars κ, φ, γ, and $\delta_1, \delta_2, \dots, \delta_m$ were chosen such that the interior of the feasible domain $\mathcal{S}$ is guaranteed to be nonempty, and such that the functions f and g are guaranteed not to have a zero in $\mathcal{S}$. Finally, we have run experiments for problems (23) with values of n ranging from $n = 50$ to $n = 500$ and values of m ranging from $m = 10$ to $m = 100$. Note that the constraints in (23) are nonlinear, and therefore, adaptations of the simplex method for solving problem (3) with data (23) would be rather complicated.

In Figure 3, we plot the function q for a typical example (23) with $n = 200$ and $m = 20$. Each '$*$' marks a point r at which the method has evaluated the function q in order to be able to guarantee that the final iterate is indeed an approximate global minimizer. Thus, each '$*$' stands for the application of an interior-point method to solve a convex problem of the form (4). Since for each '$*$' the interior-point method can be 'warm-started' using as starting points some convex combination of almost final iterates of two neighboring problems, the overall number of interior-point iterations for each '$*$' was less than eight in the average. The curve $q(r)$ is of course not known, in general. (Here, it is plotted merely for illustration; its values were determined by solving a convex problem of the form (4) for some 200 evenly spaced values of r.)

In Figure 4, we show a detailed enlargement of the points generated by Algorithm 4 near the global minimizer of a problem with $n = 100$ and $m = 20$. The plot shows that the distance between support points r on the right of the minimizer is much smaller than to the left, indicating that in this particular case, the Lipschitz bound (16) provides a much more accurate approximation to $q(r)$ than (18). Thus, the algorithm did not evaluate a further refinement for the points on the left of the minimizer.

In Table 1, we report the number of iterations taken by our Matlab implementation to solve problem (23) with $m = 20$ convex quadratic constraints and different dimensions n. The stopping criterion for these examples was chosen such that

$$q(r^{\text{final}}) - q(r^{\text{opt}}) \leqslant 10^{-4} \left(q(r^1) - q(r^{\text{opt}}) \right)$$

is guaranteed. Here, $q(r^{\text{opt}})$ is the unknown global optimum of (23). The numerical results are intended to provide a first rough estimate on the dependence of our algorithm on the dimension n of the space. We stress that the numbers of Newton steps and Hessian evaluations in Table 1 could be further reduced by a more soph-

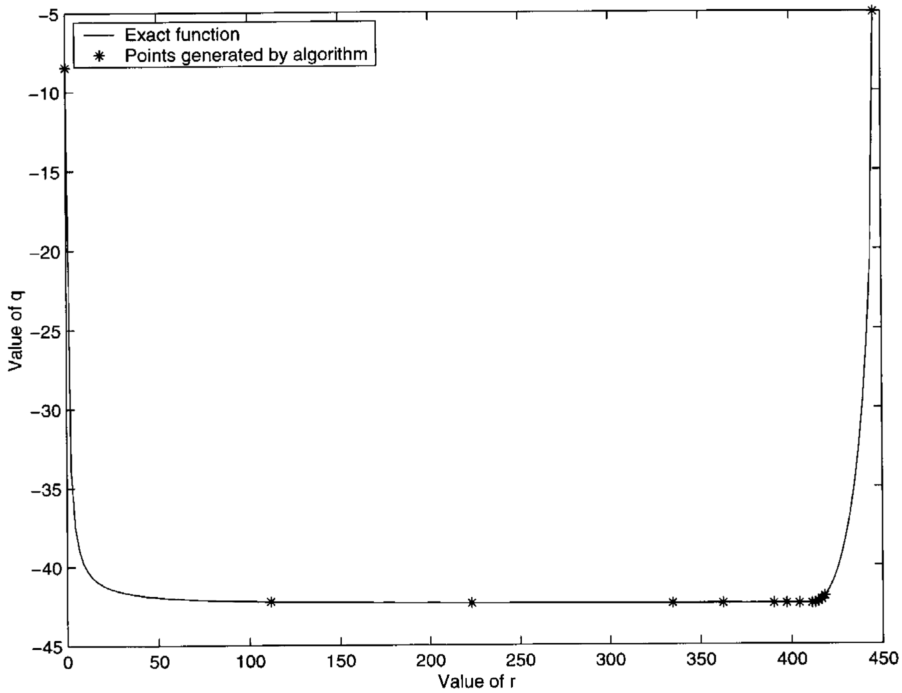

Figure 3. The function $q(r)$ for a random example, $n = 200$, $m = 20$.

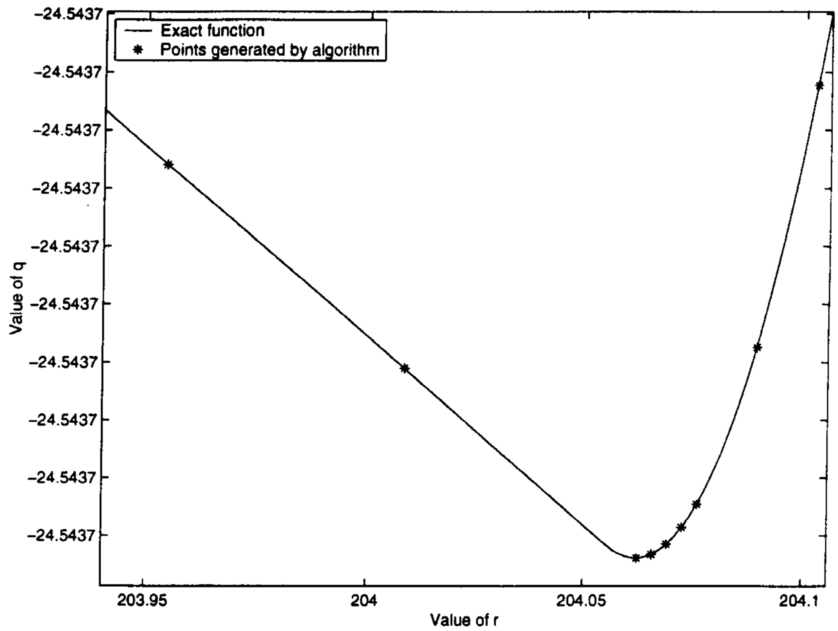

Figure 4. The function $q(r)$ near the global minimum for a random example, $n = 100$, $m = 20$.

Table 1. Iteration numbers.

Dimension	$n = 50$	$n = 100$	$n = 200$	$n = 500$
# of r's	20	17	14	14
# of IP iterations	407	384	299	326
# of Hessians	644	584	478	548
# of Newton steps	1277	1134	980	1135

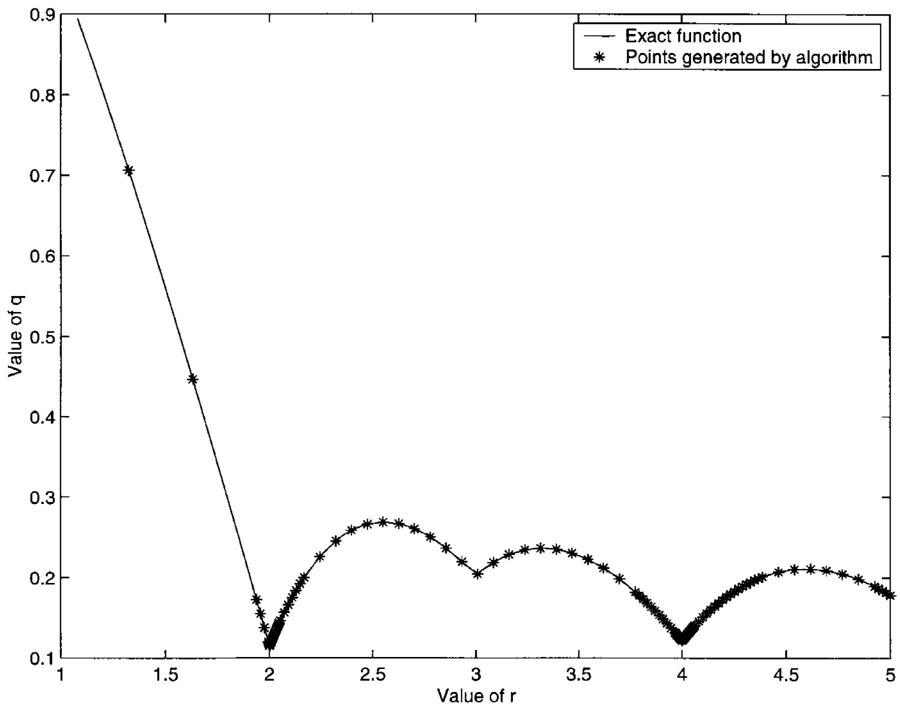

Figure 5. The function $q(r)$ for a case with several local minima.

isticated implementation. The number of Newton steps given in Table 1 refers to the sum of exact and inexact Newton steps. For inexact Newton steps, the Hessian matrix of a previous Newton step has been used in place of the current Hessian matrix. The overall computational effort is dominated by the number of Hessian evaluations. The number of r's refers to the number of support points r at which $q(r)$ was evaluated.

The random examples presented above exhibited only one local minimizer of the function q, as in Figure 3. We therefore constructed some small problems in three dimensions in such a way that there were several local minimizers at integer values of r. If two or more of the local minimizers have nearly the same value $q(r)$, the method refines about both minimizers until the global minimizer has been identified. The plot in Figure 5 shows such a 'worst-case' behavior where the method takes a large number of steps before identifying a point near $r = 2$ as an ϵ-global minimizer. If the stopping tolerance ϵ is decreased further, then only the bounds near $r = 2$ are refined to increase the accuracy of the global minimizer.

It is needless to say that Algorithm 4 does not lend itself to solving the knapsack problem of Section 2.2. The structure of the knapsack problem is not exploited by Algorithm 4, and the Lipschitz bounds (16) and (18) are too weak to provide a sufficiently sharp lower estimate for the function q for an interval of length more than one. Hence Algorithm 4 is at least as expensive as enumerating all possible integer solutions. While the knapsack problem represents an example for which Algorithm 4 is not suitable, we believe that most applications have a structure more similar to the random problems above for which Algorithm 4 provides a reliable and reasonably fast method for identifying the global minimum.

5. Conclusions

We considered the sum-of-ratios problem in $\mathbb{R}^n$ where the sum of a convex function and p convex-concave fractions is minimized subject to convex constraints. We proposed an approach to transform this problem to the problem of minimizing a suitably defined function q of p variables. The function q can be evaluated by using an interior-point method for convex minimization. We established Lipschitz bounds for q that can also be evaluated numerically by using an interior-point method. Based on these bounds, a method was derived to find an ϵ-approximation to the global minimizer of the sum-of-ratios problem.

We presented numerical experiments with the proposed algorithm for the case of minimizing the sum of a convex function and of $p = 1$ convex-concave fraction. An implementation of the algorithm for the case $p \geqslant 2$ will be described elsewhere.

Acknowledgements

We would like to thank Siegfried Schaible for bringing the sum-of-ratio problem to our attention and for providing us with a copy of the technical report (Schaible, 1996). The authors are grateful to the referees and the editor for their constructive comments that helped us to improve the presentation of the paper.

References

Aurenhammer, F. (1991), Voronoi diagrams – a survey of a fundamental geometric data structure, *ACM Computing Surveys* 23: 345–405.

Cambini, A., Martein, L. and Schaible, S. (1989), On maximizing a sum of ratios, *J. Information and Optimization Sciences* 10: 65–79.

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

Chen, D.Z., Daescu, O., Dai, Y., Katoh, N., Wu, X. and Xu, J. (1998), Efficient algorithms and implementations for optimizing the sum of linear fractional functions, with applications, Technical Report, Department of Computer Science, University of Notre Dame, Notre Dame, IN.

Falk, J.E. and Palocsay, S.W. (1992), Optimizing the sum of linear fractional functions, in: C.A. Floudas and P.M. Pardalos (eds.), *Recent Advances in Global Optimization*, Princeton University Press, Princeton, NJ, pp. 221–258.

Fortune, S.J. (1997), Voronoi diagrams and Delaunay triangulations, in: J.E. Goodman and J. O'Rourke (eds.), *Handbook of Discrete and Computational Geometry*, CRC Press, Boca Raton, FL, pp. 377–388.

Freund, R.W. and Jarre, F. (1994), An interior-point method for fractional programs with convex constraints, *Mathematical Programming* 67: 407–440.

Freund, R.W. and Jarre, F. (1995), An interior-point method for multifractional programs with convex constraints, *J. Optimization Theory and Applications* 85: 125–161.

Freund, R.W., Jarre, F. and Schaible, S. (1996), On self-concordant barrier functions for conic hulls and fractional programming, *Mathematical Programming* 74: 237–246.

Garey, M.R. and Johnson, D.S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, NY.

Hirche, J. (1985), Some remarks on generalized convexity of sums and products, *Zeitschrift für Angewandte Mathematik und Mechanik* 65: 62–63.

Hiriart-Urruty, J.-B. and Lemarechal, C. (1993), *Convex Analysis and Minimization Algorithms I*, Grundlehren der Mathematischen Wissenschaften 305, Springer-Verlag, Berlin, Germany.

Jarre, F. and Saunders, M.A. (1995), A practical interior-point method for convex programming, *SIAM J. Optimization* 5: 149–171.

Konno, H. and Kuno, T. (1990), Generalized linear multiplicative and fractional programming, *Annals of Operations Research* 25: 147–161.

Konno, H. and Yamashita, H. (1998), Minimization of the sum and the product of several linear fractional functions over a polytope, Manuscript, Department of Industrial Engineering and Management, Tokyo Institute of Technology, Tokyo, Japan.

Nemirovskii, A. (1996), On polynomiality of the method of analytic centers for fractional problems, *Mathematical Programming* 73: 175–198.

Nesterov, Y. and Nemirovskii, A. (1994), *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Studies in Applied Mathematics Vol. 13, SIAM, Philadelphia, PA.

Ritter, K. (1967), A parametric method for solving certain nonconcave maximization problems, *J. Computer and System Sciences* 1: 44–54.

Schaible, S. (1984), Simultaneous optimization of absolute and relative terms, *Zeitschrift für Angewandte Mathematik und Mechanik* 64: 363–364.

Schaible, S. (1995), Fractional programming, in: R. Horst and P.M. Pardalos (eds.), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 495–608.

Schaible, S. (1996), Fractional programming with sums of ratios, Working Paper Series No. 96-04, Graduate School of Management, University of California, Riverside, CA.