Global Optimization of Fractional Programs

P. M. PARDALOS[1] and A. T. PHILLIPS[2]
[1]*Computer Science Department, The Pennsylvania State University, University Park, PA 16802, U.S.A.*; [2]*Computer Science Department, United States Naval Academy, Annapolis, MD 21402, U.S.A.*

(Received: 19 March 1990; accepted: 11 March 1991)

Abstract. Dinkelbach's global optimization approach for finding the global maximum of the fractional programming problem is discussed. Based on this idea, a modified algorithm is presented which provides both upper and lower bounds at each iteration. The convergence of the lower and upper bounds to the global maximum function value is shown to be superlinear. In addition, the special case of fractional programming when the ratio involves only linear or quadratic terms is considered. In this case, the algorithm is guaranteed to find the global maximum to within any specified tolerance, regardless of the definiteness of the quadratic form.

Key words. Global optimization, fractional programming.

1. Introduction

Given two continuous functions $f: \boldsymbol{R}^n \to \boldsymbol{R}$ and $g: \boldsymbol{R}^n \to \boldsymbol{R}$ defined on a polyhedral set $S \subset \boldsymbol{R}^n$ such that $g(x) > 0$ for all $x \in S$, the fractional programming problem is to find some point x^* which satisfies

$$\frac{f(x^*)}{g(x^*)} = \max_{x \in S} \frac{f(x)}{g(x)} . \tag{P}$$

Applications and algorithms for fractional programs have been treated in considerable detail since the early work of Isbell and Marlow (1956). Included among the many applications are portfolio selection, stock cutting, game theory, and numerous decision problems in management science. See Grunspan (1971) for work known to up to 1971 and Schaible (1981), Schaible and Ibaraki (1983), and Avriel *et al.* (1988) for the most recent surveys.

If $f(x)$ is concave and non-negative, and $g(x)$ and S are convex (and S is bounded), then problem (P) is called a concave-convex fractional program. Schaible (1976a) showed that such problems can be solved by a single concave problem using a simple variable transformation. This provides an efficient approach for solving a limited class of fractional programming problems. Unfortunately, even in some of the simplest cases (for example when $f(x)$ and $g(x)$ are quadratic) a new constraint, which may be nonlinear, must be added (to the transformed feasible region), and the transformed problem becomes very difficult to solve. In addition, if the problem is not concave-convex initially, then the transformation does not even necessarily yield a concave problem. In fact, in the most general case, problem (P) may have many local maxima which are different from the optimal one, and hence determining the global maximum is very difficult (i.e. NP-hard).

Another more recent method is to replace the nonlinear functions by suitable linear underestimators and then obtain the global optimum by a vertex ranking procedure. This method, due to Pardalos (1986), is applicable only when $f(x)$ is a convex quadratic function and $g(x)$ is linear (hence the ratio is quasiconvex).

The last well-known approach, and one of the oldest, is to consider the global optimization problem

$$\max_{x \in S} (f(x) - \lambda g(x)) \tag{GP}$$

where $\lambda \in \boldsymbol{R}$ is a constant. This "parametric" approach, which was first proposed by Dinkelbach (1967), generates a sequence of values λ_i that converges to the global optimum function value (Schaible, 1976b). This method has since then been applied to many specific types of fractional programs including the concave-convex type, but very little work has been done to solve fractional programs where the ratio of two concave, two convex, or the ratio of a convex and a concave function is to be maximized. In addition, this method does not provide a sequence of improving upper bounds, and hence even though the sequence λ_i may be converging to the global optimum function value, no bound on the error is available at any iteration.

The method presented in this paper improves Dinkelbach's algorithm by providing a means for obtaining a sequence of improving upper bounds which, along with the corresponding sequence of improving lower bounds, will provide a bound on the error at each iteration of the solution procedure. In addition, both the sequence of lower bounds and the sequence of upper bounds are shown to converge to the global optimum function value at a "superlinear" rate. The new algorithm is also appropriate for the class of quadratic fractional programs (i.e. one or both of $f(x)$ and $g(x)$ are quadratic) where the ratio may involve concave, convex, or even indefinite terms. It combines Dinkelbach's approach with a method guaranteed to solve linearly constrained quadratic programming problems regardless of the definiteness of the quadratic from (Phillips and Rosen, 1990a).

Two algorithms which are similar to the one presented here are given by Schaible (1976b) and Ibaraki (1983). Schaible's method first computes a sequence of improving upper and lower bounds using an efficient section method. Dinkelbach's algorithm is then started as soon as the section method achieves a set of bounds that differ by some prespecified tolerance. The algorithm presented here differs from Schaible's method in that the upper and lower bounds are continously improving throughout the procedure. Nevertheless, in both algorithms the sequence of upper and lower bounds converges superlinearly.

Likewise, Ibaraki (1983) presents a variety of related algorithms which also provide upper and lower bounds. These algorithms combine Dinkelbach's approach with various search techniques (e.g., Newton, binary, modified binary). The result is a set of related algorithms with convergence rates that vary depending on the search technique employed. Ibaraki (1983) also provides a collection of computational results for the fractional knapsack problem and quadratic fractional programs.

2. Problem Formulation and Mathematical Properties

The fundamental result which relates the global optimization approach (GP) to the general fractional programming problem (P) is

THEOREM 1 (Dinkelback 1967). *x^* solves the fractional programming problem (P) iff x^* solves the global optimization problem (GP) with constant $\lambda^* = f(x^*)/g(x^*)$.*

Dinkelbach's original iterative algorithm is based on the result of this theorem and can be described as

Dinkelbach(S, f, g):
1. Select some $x^{(0)} \in S$. Set $\lambda^{(0)} := f(x^{(0)})/g(x^{(0)})$ and $k := 0$.
2. Solve the constrained global optimization problem
$$\max_{x \in S}(f(x) - \lambda^{(k)} g(x))$$
to get the optimal solution point $x^{(k+1)}$.
3. If $f(x^{(k+1)}) - \lambda^{(k)} g(x^{(k+1)}) = 0$, then set $x^* := x^{(k+1)}$ and $\lambda^* := \lambda^{(k)}$ and stop.
4. If $f(x^{(k+1)}) - \lambda^{(k)} g(x^{(k+1)}) > 0$, then set $\lambda^{(k+1)} := f(x^{(k+1)})/g(x^{(k+1)})$, $k := k + 1$ and go to step 2.

Clearly, the efficiency of this algorithm depends on the number of times the constrained global optimization problem must be solved, and on the time spent solving it during each iteration. Also note that a test of the form $f(x^{(k+1)}) - \lambda^{(k)} g(x^{(k+1)}) < 0$ is not necessary since, for any fixed k,

$$f(x^{(k+1)}) - \lambda^{(k)} g(x^{(k+1)}) = \max_{x \in S}(f(x) - \lambda^{(k)} g(x)) \geq f(x^{(k)}) - \lambda^{(k)} g(x^{(k)}) = 0 .$$

Now consider the function $M(\lambda)$ defined as

$$M(\lambda) = \max_{x \in S}(f(x) - \lambda g(x)) .$$

The function $M(\lambda)$ has two interesting properties that will be important in proving convergence of upper and lower bounds to λ^* and in determining the rate of this convergence. The first of these properties is that for any lower bound λ of λ^*, $M(\lambda)$ is positive, and for any upper bound λ of λ^*, $M(\lambda)$ is negative. Secondly, the function $M(\lambda)$ is convex. That is,

LEMMA 1 (Dinkelbach 1967).
(1) *$M(\lambda) > 0$ for all $\lambda < \lambda^*$, and $M(\lambda) < 0$ for all $\lambda > \lambda^*$, and*
(2) *$M(\lambda)$ is convex.*

The sequence of iterates $\lambda^{(0)}, \lambda^{(1)}, \ldots$ generated by algorithm "Dinkelbach-(S, f, g)" is strictly monotone increasing, and satisfy $M(\lambda^{(i)}) > 0$ for $i = 0, 1, \ldots$ (Dinkelbach 1967). Hence, by Lemma 1 they provide a strictly monotone increasing sequence of lower bounds for λ^*.

3. Lower Bounds and Convergence Rates

A sequence of iterates $\alpha^{(0)}, \alpha^{(1)}, \ldots$ is said to converge *superlinearly* to α^* if

$$\lim_{n \to \infty} \frac{|\alpha^* - \alpha^{(n+1)}|}{\theta^{n+1}} = 0$$

for all $\theta \in (0, 1]$. In order to demonstrate that the sequence of lower bounds $\lambda^{(i)}$ converges superlinearly to $\lambda^* \equiv f(x^*)/g(x^*)$ where x^* is any optimal solution for (P), consider the following

LEMMA 2 (Schaible 1979). *If $x^* \in S$ is any optimal solution for (P) and x' solves $M(\lambda')$ and $\lambda' < \lambda^*$, then*

$$\lambda^* - \frac{f(x')}{g(x')} \leq (\lambda^* - \lambda')\left(1 - \frac{g(x^*)}{g(x')}\right)$$

where $0 \leq 1 - g(x^*)/g(x') < 1$.
From Lemma 2, with $\lambda' = \lambda^{(n)}$ and $x' = x^{(n+1)}$ (since x' is the solution to $M(\lambda')$ and the algorithm "Dinkelbach(S, f, g)" denotes the solution of $M(\lambda^{(n)})$ by $x^{(n+1)}$) we get the error

$$(\lambda^* - \lambda^{(n+1)}) \leq (\lambda^* - \lambda^{(n)})\left(1 - \frac{g(x^*)}{g(x^{(n+1)})}\right).$$

Now let $\varepsilon_i = (1 - g(x^*)/g(x^{(i)}))$ for $i = 1, 2, \ldots$, where (by Lemma 2) $0 \leq \varepsilon_i < 1$. Then it can be shown that $\varepsilon_1 \geq \varepsilon_2 \geq \cdots \geq 0$ (Schaible, 1976b). In addition, from Lemma 2 with $x' = x^{(1)}$ and $\lambda' = \lambda^{(0)}$, it is clear that $\varepsilon_1 < 1$. In order to prove that the sequence of iterates $\lambda^{(0)}, \lambda^{(1)}, \ldots$ converges to λ^* and that rate of convergence is in fact superlinear, we need the following important result.

LEMMA 3 (Schaible 1973). *There exists an $x^* \in S$, an optimal solution for (P), such that*

$$\lim_{n \to \infty} \varepsilon_n = 0.$$

Thus, there is some $x^* \in S$, an optimal solution for (P), such that the sequence $\varepsilon_1, \varepsilon_2, \ldots$ converges to 0, and (by induction on the error result given above)

$$(\lambda^* - \lambda^{(n+1)}) \leq (\lambda^* - \lambda^{(0)}) \prod_{i=1}^{n+1} \left(1 - \frac{g(x^*)}{g(x^{(i)})}\right) \leq (\lambda^* - \lambda^{(0)}) \prod_{i=1}^{n+1} \varepsilon_i.$$

Hence, for any $\theta \in (0, 1]$, there exists some constant integer $n_\theta \geqslant 0$ such that for all $i \geqslant n_\theta$ we get $\varepsilon_i < \theta$. Defining the constant

$$\beta_\theta = \prod_{i=1}^{n_\theta - 1} \left(\frac{\varepsilon_i}{\theta}\right)$$

we get that

$$\frac{(\lambda^* - \lambda^{(n+1)})}{\theta^{n+1}} \leqslant (\lambda^* - \lambda^{(0)})\beta_\theta \prod_{i=n_\theta}^{n+1} \left(\frac{\varepsilon_i}{\theta}\right)$$

and therefore

$$\lim_{n\to\infty} \frac{(\lambda^* - \lambda^{(n+1)})}{\theta^{n+1}} = 0 .$$

That is, the sequence of lower bounds $\lambda^{(n+1)}$ converges to $\lambda^* \equiv f(x^*)/g(x^*)$ and does so at a superlinear rate (see also Schaible, 1976b).

4. Upper Bounds and Convergence Rates

As it now stands, the algorithm "Dinkelbach(S, f, g)" does not provide upper bounds on the global optimum function value $\lambda^* \equiv f(x^*)/g(x^*)$. One way to obtain an *initial* upper bound is to solve the following two problems:

$$\max_{x \in S} f(x)$$

to get the optimal solution $f(x')$, and

$$\min_{x \in S} g(x)$$

to get the optimal solution $g(x'')$. Then an initial upper bound is clearly given by $\gamma^{(-1)} \equiv f(x')/g(x'')$. In fact, according to Lemma 1 (part 1), any $\gamma \in \boldsymbol{R}$ satisfying $M(\gamma) < 0$ would also be an upper bound. Hence, if we define

$$\gamma^{(n)} \equiv \gamma^{(n-1)} - M(\gamma^{(n-1)})\left(\frac{\gamma^{(n-1)} - \lambda^{(n)}}{M(\gamma^{(n-1)}) - M(\lambda^{(n)})}\right)$$

where $\lambda^{(n)}$ is the most recent lower bound of λ^* and $\gamma^{(n-1)}$ is the most recent upper bound of λ^*, then the new upper bound is given by $\gamma^{(n)}$. Figure 1 illustrates that $\gamma^{(n)}$ is just the root of the line segment joining the points $(\lambda^{(n)}, M(\lambda^{(n)}))$ and $(\gamma^{(n-1)}, M(\gamma^{(n-1)}))$.

This leads to an important modification of the algorithm "Dinkelbach(S, f, g)":

Fract(S, f, g, δ):
1. Select some $x^{(0)} \in S$. Set $\lambda^{(0)} := f(x^{(0)})/g(x^{(0)})$.
2. Solve the constrained global optimization problems

$$\max_{x \in S} f(x)$$

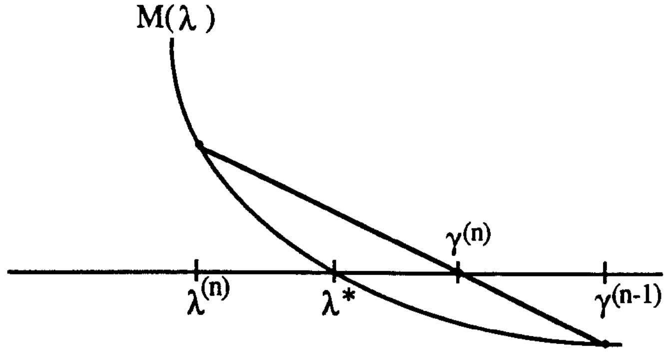

Fig. 1.

and

$$\min_{x \in S} g(x)$$

to get the optimal function values $f(x')$ and $g(x'')$, respectively. Set $\gamma^{(-1)} := f(x')/g(x'')$ and $k := 0$. If $\gamma^{(-1)} - \lambda^{(0)} \leqslant \delta$, then set $\lambda^* := \lambda^{(0)}$ and $x^* := x^{(0)}$ and stop.

3. Solve the constrained global optimization problem

$$M(\lambda^{(k)}) \equiv \max_{x \in S}(f(x) - \lambda^{(k)} g(x))$$

to get the optimal solution point $x^{(k+1)}$.

4. If $M(\lambda^{(k)}) = 0$, then set $x^* := x^{(k+1)}$ and $\lambda^* := \lambda^{(k)}$ and stop.
5. Solve the constrained global optimization problem

$$M(\gamma^{(k-1)}) \equiv \max_{x \in S}(f(x) - \gamma^{(k-1)} g(x))$$

to get the optimal solution point $y^{(k)}$.

6. If $M(\gamma^{(k-1)}) = 0$, then set $x^* := y^{(k)}$ and $\lambda^* := \gamma^{(k-1)}$ and stop.
7. Set

$$\gamma^{(k)} := \gamma^{(k-1)} - M(\gamma^{(k-1)}) \left(\frac{\gamma^{(k-1)} - \lambda^{(k)}}{M(\gamma^{(k-1)}) - M(\lambda^{(k)})} \right).$$

8. If $\gamma^{(k)} - \lambda^{(k)} \leqslant \delta$, then set $\lambda^* := \lambda^{(k)}$ and $x^* := x^{(k+1)}$ and stop.
9. Set $\lambda^{(k+1)} := f(x^{(k+1)})/g(x^{(k+1)})$, $k := k+1$ and go to step 3.

Note that the parameter $\delta \geqslant 0$ is a user supplied stopping tolerance. The following theorem shows that the sequence of iterates $\gamma^{(-1)}, \gamma^{(0)}, \gamma^{(1)}, \ldots$ is, in fact, a sequence of upper bounds on λ^*, and that the sequence is strictly monotonically decreasing.

THEOREM 2. $\lambda^* \leqslant \gamma^{(i+1)} < \gamma^{(i)}$ *for* $i = -1, 0, 1, \ldots$

Proof. Clearly $\lambda^* \leqslant \gamma^{(-1)}$. Let $n \geqslant 0$ and consider the line segment $L(\lambda)$ joining the points $(\lambda^{(n)}, M(\lambda^{(n)}))$ and $(\gamma^{(n-1)}, M(\gamma^{(n-1)}))$. Note that if $M(\lambda^{(n)}) = 0$, then

$\lambda^* = \lambda^{(n)}$ and $\gamma^{(n)}$ would not be computed. Likewise, if $M(\gamma^{(n-1)}) = 0$ then $\lambda^* = \gamma^{(n-1)}$ and $\gamma^{(n)}$ would not get computed. Thus, without loss of generality, assume that $M(\lambda^{(n)}) \neq 0$ and $M(\gamma^{(n-1)}) \neq 0$. By induction, if $\lambda^{(n)} < \lambda^*$ and $\gamma^{(n-1)} > \lambda^*$ then, by Lemma 1 (part 1), $L(\lambda^{(n)}) = M(\lambda^{(n)}) > 0$ and $L(\gamma^{(n-1)}) = M(\gamma^{(n-1)}) < 0$ so that the root $\gamma^{(n)}$ of $L(\lambda)$ satisfies $\lambda^{(n)} < \gamma^{(n)} < \gamma^{(n-1)}$. Hence, $\gamma^{(n)} < \gamma^{(n-1)}$ so that the sequence is strictly monotonically decreasing. In addition, $\gamma^{(n)} = q\lambda^{(n)} + (1-q)\gamma^{(n-1)}$ for some $q \in (0, 1)$, and since $M(\lambda)$ is convex (Lemma 1 (part 2)), we get

$$\begin{aligned} M(\gamma^{(n)}) &= M(q\lambda^{(n)} + (1-q)\gamma^{(n-1)}) \leq qM(\lambda^{(n)}) + (1-q)M(\gamma^{(n-1)}) \\ &= qL(\lambda^{(n)}) + (1-q)L(\gamma^{(n-1)}) \\ &= L(q\lambda^{(n)} + (1-q)\gamma^{(n-1)}) = L(\gamma^{(n)}) = 0 \,. \end{aligned}$$

That is, $M(\gamma^{(n)}) \leq 0$ so that, by Lemma 1 (part 1), $\lambda^* \leq \gamma^{(n)}$. □

To show that the sequence of upper bounds $\gamma^{(i)}$ converges to $\lambda^* \equiv f(x^*)/g(x^*)$, and that this convergence is superlinear, consider the definition of the $(n+1)^{\text{st}}$ iterate (assuming, of course, that the solution has not yet been found, i.e. $M(\lambda^{(n+1)}) \neq 0$ and $M(\gamma^{(n)}) \neq 0$)

$$\gamma^{(n+1)} = \gamma^{(n)} - M(\gamma^{(n)})\left(\frac{\gamma^{(n)} - \lambda^{(n+1)}}{M(\gamma^{(n)}) - M(\lambda^{(n+1)})}\right).$$

After some algebraic manipulation, it can be shown that

$$\begin{aligned} (\lambda^* - \gamma^{(n+1)}) &= (\lambda^* - \gamma^{(n)}) + M(\gamma^{(n)})\left(\frac{\gamma^{(n)} - \lambda^{(n+1)}}{M(\gamma^{(n)}) - M(\lambda^{(n+1)})}\right) \\ &= -(\lambda^* - \lambda^{(n+1)})(\lambda^* - \gamma^{(n)}) \frac{M''(\xi_{n+1})}{2M'(\zeta_{n+1})} \end{aligned}$$

for some $\zeta_{n+1}, \xi_{n+1} \in [\lambda^{(n+1)}, \gamma^{(n)}]$. Now, let

$$\kappa = \frac{\max\limits_{\xi \in I} |M''(\xi)|}{2 \min\limits_{\zeta \in I} |M'(\zeta)|}$$

where $I = [\lambda^{(0)}, \gamma^{(-1)}]$. Note that $I_{n+1} \equiv [\lambda^{(n+1)}, \gamma^{(n)}] \subset I$ for $n = -1, 0, 1, \ldots$ so that κ is an upper bound on

$$\frac{|M''(\xi_{n+1})|}{2|M'(\zeta_{n+1})|} \quad \text{for } n = -1, 0, 1, \ldots$$

An upper bound on the error at the $(n+1)^{\text{st}}$ iteration can now be written as

$$|\lambda^* - \gamma^{(n+1)}| \leq |\lambda^* - \lambda^{(n+1)}||\lambda^* - \gamma^{(n)}|\kappa \,.$$

Hence, by induction we get that

$$|\lambda^* - \gamma^{(n+1)}| \leq \kappa^{n+2}|\lambda^* - \gamma^{(-1)}| \prod_{i=0}^{n+1} |\lambda^* - \lambda^{(i)}|$$

and recalling (from the analysis of the convergence of the sequence of lower bounds $\lambda^{(i+1)}$ to λ^*) that

$$(\lambda^* - \lambda^{(i+1)}) \leq (\lambda^* - \lambda^{(i)})\varepsilon_{i+1}$$

where $\varepsilon_i = (1 - g(x^*)/g(x^{(i)}))$ and where $0 \leq \varepsilon_i < 1$, then

$$|\lambda^* - \gamma^{(n+1)}| \leq \kappa^{n+2}|\lambda^* - \gamma^{(-1)}||\lambda^* - \lambda^{(0)}|^{n+2} \prod_{i=1}^{n+1} \varepsilon_i^{n-i+2} .$$

Again, by Lemma 3, the sequence $\varepsilon_1, \varepsilon_2, \ldots$ converges to 0. Hence, for any $\theta \in (0, 1]$, there exists some constant integer $n_\theta \geq 0$ such that for all $i \geq n_\theta$ we get $(\kappa|\lambda^* - \lambda^{(0)}|\varepsilon_i) < \theta$. Thus, by defining the constant

$$\alpha_\theta = \kappa|\lambda^* - \lambda^{(0)}| \prod_{i=1}^{n_\theta - 1} \left(\frac{\kappa|\lambda^* - \lambda^{(0)}|\varepsilon_i^{n-i+2}}{\theta}\right)$$

we get that

$$\frac{|\lambda^* - \gamma^{(n+1)}|}{\theta^{n+1}} \leq |\lambda^* - \gamma^{(-1)}|\alpha_\theta \prod_{i=n_\theta}^{n+1} \left(\frac{\kappa|\lambda^* - \lambda^{(0)}|\varepsilon_i^{n-i+2}}{\theta}\right).$$

Hence,

$$\lim_{n\to\infty} \frac{|\lambda^* - \gamma^{(n+1)}|}{\theta^{n+1}} = 0$$

so that the sequence $\gamma^{(n+1)}$ converges to λ^*, and does so at a superlinear rate.

5. Special Cases

If the feasible set S is polyhedral and the functions $f(x)$ and $g(x)$ are either linear or quadratic, then the algorithm solves a sequence of linear or quadratic programs, respectively. In particular, if $f(x) = c^t x$ and $g(x) = d^t x$ then the algorithm solves the sequence of linear programs

$$\max_{x\in S}(c - \lambda^{(k)}d)^t x .$$

If both $f(x)$ and $g(x)$ are quadratic, i.e., $f(x) = (1/2)x^t Qx + c^t x$ and $g(x) = (1/2)x^t Px + d^t x$, then the algorithm solves the sequence of quadratic programs

$$\max_{x\in S} \frac{1}{2} x^t(Q - \lambda^{(k)}P)x + (c - \lambda^{(k)}d)^t x .$$

Notice that the matrix $(Q - \lambda^{(k)}P)$ may be indefinite, in which case the algorithm is required to find the global maximum of a linearly constrained indefinite quadratic function. Even though this is an NP-hard problem (e.g., when $(Q\text{-}\lambda^{(k)}P)$ is positive definite), the method developed by Phillips and Rosen (1990a) is guaranteed to find an ε-approximate global maximum (i.e., the relative error is no larger than ε) for any specified $\varepsilon > 0$.

Furthermore, if $f(x)$ and $g(x)$ are such that $f(x) - \lambda\, g(x)$ is only "partially separable", then the method developed by Phillips and Rosen (1990b) can be used to find an ε-approximate global maximum for any specified $\varepsilon > 0$. That is, the method of Phillips and Rosen (1990b) is guaranteed to find solutions to the sequence of subproblems

$$M(\lambda^{(k)}) \equiv \max_{x \in S}(f(x) - \lambda^{(k)} g(x))$$

and

$$M(\gamma^{(k-1)}) \equiv \max_{x \in S}(f(x) - \gamma^{(k-1)} g(x))$$

if x can be partitioned into two components $x = (w, z)$ such that $f(x) - \kappa g(x)$ (where the constant $\kappa = \lambda^{(k)}$ or $\gamma^{(k-1)}$) can be written in the form $\varphi(w) + \psi(z)$ where $\varphi(w)$ is a separable convex function of w and $\psi(z)$ is a concave (but not necessarily separable) function of z. The applicability of these methods to the solution of these subproblems greatly extends the class of fractional programming problems that can be solved in practice. Computational results on a class of fractional programming problems which can be put into partially separable form (which includes the class of quadratic functions) are forthcoming.

Acknowledgements

This research was supported in part by the National Science Foundation grant ASC-8902036, the Air Force Office of Scientific Research grant AFOSR 89-0031, by the Minnesota Supercomputer Institute, and Cray Research, Inc.

References

Avriel, M., W. E. Diewert, S. Schaible, and I. Zang (1988), *Generalized Concavity*, New York: Plenum Press.

Dinkelbach, W. (1967), On Nonlinear Fractional Programming, *Management Science* **13**(7): 492–498.

Grunspan, M. (1971), Fractional Programming: A Survey, Technical Report 50, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL.

Ibaraki, T. (1983), Parametric Approaches to Fractional Programs, *Mathematical Programming* **26**: 345–362.

Isbell, J. R. and W. H. Marlow (1956), Attrition Games, *Naval Research Logistics Quarterly* **3**: 71–93.

Pardalos, P. M. (1986), An Algorithm for a Class of Nonlinear Fractional Problems Using Ranking of the Vertices, *BIT* **26**: 392–395.

Phillips, A. T. and J. B. Rosen (1990a), Guaranteed ε-Approximate Solution for Indefinite Quadratic Global Minimization, *Naval Research Logistics* **37**: 499–514.

Phillips, A. T. and J. B. Rosen (1990b), A Parallel Algorithm for Partially Separable Non-convex Global Minimization, *Annals of Operations Research* **25**: 101–118.

Schaible, S. (1973), Fractional Programming: Transformations, Duality, and Algorithmic Aspects, Technical Report 73-9, Department of Operations Research, Stanford University, Stanford, CA.

Schaible, S. (1976a), Fractional Programming I, Duality, *Management Science* **22**(8): 858–867.

Schaible, S. (1976b), Fractional Programming II, On Dinkelbach's Algorithm, *Management Science* **22**(8); 868–873.

Schaible, S. (1981), Fractional Programming: Applications and Algorithms, *European Journal of Operational Research* **7**: 111–120.

Schaible, S. and T. Ibaraki (1983), Fractional Programming, *European Journal of Operational Research* **12**: 325–338.