An. Şt. Univ. Ovidius Constanţa Vol. 11(2), 2003, 87–92

A PARAMETRIC STUDY FOR SOLVING NONLINEAR FRACTIONAL PROBLEMS

Antoneta Jeflea

1. Introduction

The non-linear fractional fractional programming problem, i.e. the minimization of a fraction of two functions subject to given conditions, arises in various decision making situations; for example linear fractional programming is used in fields of game theory, network flows; the quadratic fractional programming problem is used on field production planning and inventories.

There are different solution algorithms for determing the optimal solution of particular kinds of fractional programming problems. For example, the authors Charnes and Cooper (1962), Isbell and Marlow (1962), Martos (1964) and Wolf (1985) solve linear fractional programming. Integer linear fractional programming has been solved by Rajendra (1993), Seshan and Tibekar (1980), Chandra and Chandramoham (1980), etc. Swarup (1965) gives an algorthm for solving quadratic fractional programming. The case where the restrictions are linear and the objective function is the quotient of a convex function with a concave function is solved by Mangasarian (1969) using Frank and Wolfe′s algorithm (1956). Dinkelbach (1968) also considered the same objective over a convex feasible set. He solved this problem by means of the solution of a sequence of non-linear convex programming problems.

There are other fields of application where exact algorithms do not exist to solve fractional programming. An example of this is given in Gopal et al.(1991) which investigates configuration management and optimal logical network design for reconfigurable networks. They defined underlying constrained non-linear integer fractional problems and developed a heuristic technique to solve it.

In this paper several algorithms are presented to solve non-linear fractional programming based on the parametric approach to the fractional programming problem given by Dinkelbach. We prove their convergence and this opens the posibility of developing new exact algorithms or heuristic procedures for problems formulated by means of fractional programming.

This paper is organized as follows. In Section 2 we introduce the fractional problem and Dinkelbach′s algorithm and we prove the global convergence of Dinkelbach′s algorithm for the general case of the fractional programming. In Section 3, a possible extension is given for the speed-up of Dinkelbach′s algorithm. It consists of solving the subproblems inexactly. We prove the global convergence of the method under the hypothesis that the algorithm employed for the subproblem is a descent algorithm whose algorithm map is closed.

2. Dinkelbach′s algorithm.

2.1 Notation and preliminaries.

The general fractional programming can be formulated as the following problem:

(P) $\left\{\min\ \theta(x) = \frac{\Phi(x)}{\Psi(x)} : x \in X\right\}$

where X is a nonempty compact of R^n. The functions $\Phi(x)$ and $\Psi(x)$ are continuous real-valued functions of $x \in X$. Furthermore, the following assumption is also made:

$\Psi(x) > 0$ for all $x \in X$. (2.1)

θ is a continuous function of the compact set X and thus **(P)** has a solution. By Ω, we denote the set of solution of **(P).**

Jagannathan (1966) supplied theorethical insight into the relationship between non-linear fractional programming and non-linear parametric programming. He studied the relationship of the problem **(P)** with the following problem:

$(P(\delta))$ min $\{\Phi(x) - \delta\Psi(x) : x \in X\}$

and proved the following theorem:

Theorem 2.1 (Jagannathan′s theorem). Let $y \in X$, y is an optimal solution for **(P)** if and only if y is an optimal solution for

min $\{\Phi(x) - \theta(y)\Psi(x) : x \in X\}$

The problem $P(\delta)$ studied by Jagannathan has a solution for any $\delta \in R$ because X is a compact set of R^n and Φ and Ψ are continuous on X. We can define:

$f(\delta) = \min\ \{\Phi(x) - \delta\Psi(x) : x \in X\}$

Dinkelbach (1968) developed a method based on Jagannathan′s theorem for solving non-linear fractional problems where the function Ψ is concave and Φ is convex. He proved the convergence of the algorithm for this case. The original Dinkelbach′s algorithm may be stated as follows:

Step 1 Let x_1 be a feasible point of X and $\delta_1 = \theta(x_1)$. Let $k = 1$ and go to Step 2.

Step 2 (Subproblem) By means of any method of convex programming solve the following subproblem:

SUB(k):

$f(\delta_k) = \min\ \{\Phi(x) - \delta_k \Psi(x) : x \in X\}$

and denote any solution point by x_{k+1}

Step 3 If $f(\delta_k) = 0$, stop and x_k is optimal. Otherwise, set $\delta_{k+1} = \theta(x_{k+1})$ and $k = k + 1$,and go to Step 2.

2.2 Global convergence

The subsection shows that Dinkelbach′s algorithm is also valid to solve the general fractional problem. The applicability of the method is based on the possibility of solving the subproblem **SUB(k)** generated in all iterations, that are not necessarily convex programmes. We will prove that the algorithm is convergent for solving **(P).**

Let $\{x_k\}$ be a sequence of points of X, we denote by θ^k the function:

$\theta^k(x) = \Phi(x) - \theta(x_k)\Psi(x)$.

Lemma 2.1 *Let $\{x_k\}$ be a sequence of points of X. If $\theta^k(x) < 0$ holds for some $x \in X$, then $\theta(x) < \theta(x_k)$.*

Proof. The hypothesis justifies the following expression

$\theta^k(x) = \Phi(x) - \theta(x_k)\Psi(x) < 0$.

Removing $\theta(x_k)$ from the previous relation and using (2.1), we obtain $\theta(x) < \theta(x_k)$.

The basic descendent property is given by the following lemma:

Lemma 2.2 *Assume that x_k is a feasible point in **(P)**, and that x_{k+1} solves **SUB(k).** If x_k solves **SUB(k),** then x_k is optimal in **(P).** Otherwise, $\theta(x_{k+1}) < \theta(x_k)$.*

Proof. If x_k solves **SUB(k),** then Jagannathan′s theorem guaranties that x_k solves **(P).** Otherwise, as x_{k+1} solves **SUB(k).**

$\varphi^k(x_{k+1}) < \varphi^k(x_k) = \Phi(x_k) - \theta(x_k)\Psi(x_k) = \Phi(x_k) - \frac{\Phi(x_k)}{\Psi(x_k)}\Psi(x_k) = 0$

and by using Lemma 2.1 we obtain $\theta(x_{k+1}) < \theta(x_k)$.

Lemma 2.2 justifies the checking of the optimality in Step 3 of the algorithm. If the convergence is not detected after having solved **SUB(k),** the algorithm proceeds by defining the new x_{k+1}as a solution of **SUB(k).**

The basic convergence property of this algorithm is given below.

Theorem 2.2 *Dinkelbach′s algorithm either terminates in a finite number of iterations or it generates an infinite sequenceso that any accumulationpoint solves **(P).***

Proof. We prove that the map of Dinkelbach′s algorithm (which it is denoted by D) is closed on $X - \Omega$.

Let $\{x_k\}$ and $\{y_k\}$ be two sequences satisfying:

$$x_k \in X \text{ and } \lim_{k\to\infty} x_k = \bar{x} \in X - \Omega,$$

$$y_k \in D(x_k) \text{ and } \lim_{k\to\infty} y_k = \bar{y}\ . \tag{2.2}$$

We show that $\bar{y} \in D(\bar{x})$.

Using the fact that $y_k \in X$ and X is a closed set, we obtain $\bar{y} \in X$.

We define $\Gamma(x, y) = \Phi(y) - \frac{\Phi(x)}{\Psi(x)}\Phi(y)$. Let $\hat{y}$ be $\hat{y} \in D(\bar{x})$, therefore it satisfies the following inequality:

$$\Gamma(\bar{x}, \hat{y}) \leq \Gamma(\bar{x}, \bar{y}). \tag{2.3}$$

By hypothesis (2.2), y_k solves $P(\theta(x_k))$ and the following expression is verified

$$\Gamma(x_k, y_k) \leq \Gamma(x_k, \hat{y}). \tag{2.4}$$

The function $\Gamma(x, y)$ is continuous on $X \times X$ in particular at $(\bar{x}, \bar{y})$. Taking limit on both sides of expression (2.4), we obtain:

$$\Gamma(\bar{x}, \bar{y}) \leq \Gamma(\bar{x}, \hat{y}).$$

This relation joint at (2.3) guaranties that $\Gamma(\bar{x}, \hat{y}) = \Gamma(\bar{x}, \bar{y})$ and $\bar{y}$ solves $P(\theta(\bar{x}))$.

We have proved that the algorithmic mapis closed on $X - \Omega$ and that Lemma 2.2 is an algorithmic descent. These properties ensure convergence of the algorithm. Thus the proof is complete.

3 Dinkelbach′s Truncated Algorithm

Patriksson (1993) introduced the class of partial linearization methods. These solve a continuous optimization problem by means of the solving of a sequence of subproblems of optimization in the original feasible region. The solution of these subproblems defines a descent direction in all iterations.

Patriksson considered that from a practical point of view, the subproblems can not be solved exactly, and there must be a trade-off between the amount of work spent on solving the subproblem and obtaining sufficient step descent.

The idea behind the truncated algorithm is to limit the work performed on the subproblem, by limiting the number of iterations performed with a finite integer n_k. From above this numbers can either be determined a priori, or be viewed as being the consequence of the algorithm and stopping criteria chosen for the subproblems.

This strategy to speed up the partiallinearization methods has been adapted to Dinkelbach′s algorithm. in this context the subproblems SUB(k) will be solved inaccurately by means of realising n_k iterations with a descent algorithm.

It will be shown that the sequence $\{n_k\}$ may be chosen arbitrarily, with $n_k \geq 1$ $(\forall)$ k, and convergence will still be ensured under the condition that the method used for solving SUB(k) has a closed algorithmic map. The following theorem establishes the global convergence of the truncated algorithm.

Theorem 3.1. *Assume thet the algorithm used for solving the SUB(k) is a descent algorithm with closed algorithmic map on the class of problems* $P(\delta)$

and that the termination criteria chosen for SUB(k) is to realise n_k iterations , $1 \leq n_k \leq \infty$, with the algorithm. We assume that the initial guess for the SUB(k) is the point x_k. Thus the algorithm either terminates in a finite number of iterations or it generates an infinite sequence $\{x_k\}$ such that any accumulation point solves (P).

Proof. The point x_{k+1} is obtained making in $P(\delta_k)$ more than on iteration with a descent algorithm and initial guess x_k. This fact guarantees that if x_k is not an optimal solution for SUB(k) then

$\phi^k(x_{k+1}) < \phi^k(x_k) = 0$.

Otherwise, x_k is an optimal solution for SUB(k) and Jagannathan′s theorem justifies that x_k is also an optimal solution for (P). Moreover $\phi^k(x_{k+1}) = 0$ and Dinkelbach′s truncated algorithm detects the optimality of x_k.

Assume therefore that $\phi^k(x_{k+1}) < 0,\ (\forall)\ k$.

Lemma 2.1 guarantees that the sequence $\{\theta(x_k)\}$ is decreasing and monotone. It is low bounded by the optimal value of (P), thus the sequence is convergent.

$\lim\limits_{k\to\infty} \theta(x_k) = \delta^* \quad (3.1)$ and any subsequence is convergent at δ^*.

Assume that , for any positive constant S, there exists a finite integer i such that $n_k \geq S,\ (\forall)\ k \geq i$. We say that $\lim\limits_{k\to\infty} n_k = +\infty$. Thus the subproblemis solved accuratelyin the limit, and convergence is ensured by Theorem 2.2. Otherwise, let $\{y_k\}$ be a subsequence of $\{x_k\}$ satisfying $\lim\limits_{k\to\infty} y_k = y, \quad (3.2)$ therefore we will prove that y solves (P).

Since $\lim\limits_{k\to\infty} n_k \neq +\infty$, there must be an integer k^* that occurs in the sequence $\{n_k\}$,an inmfinite number of times. Choose the subsequence $\{u_k\}$ of $\{y_k\}$ corresponding to the indices. Let M be the algorithmic map defined by the composite of k^* consecutive times of the closed algorithmic map used to solve the subproblems (which we denote as A) with the function

B: $X \times [\delta_{MIN}, \delta_{MAX}] \to X \times [\delta_{MIN}, \delta_{MAX}]$

$$(x, \delta) \to (x, \theta(x))$$

where δ_{MIN} and δ_{MAX} are respectively the minimum and the maximum values of (P), i.e. $M = AA...AB$ where A is k^* times.

The mapping **A** is closed and $B(x, \delta) = (x, \theta(x))$ is a continuous function on its domain, thus the composite mapping M is closed.

We consider the subsequence of $\{x_k\}$, that we denote as $\{y_k^{'}\}$, that satisfies $(y_k^{'}, \theta(u_k)) \in M(u_k, \theta(u_k))$

Without loss of generality we can assume that the sequence $\{y_k^{'}\}$ is convergent, $\lim_{k\to\infty} y_k^{'} = y^{'}$.

$\theta(u_k)$ and $\theta(y_k^{'})$ are two subsequence of the convergent sequence $\{\theta(x_k)\}$ and using (3.1)

$$\lim_{k\to\infty} \theta(u_k) = \delta^*, \lim_{k\to\infty} \theta(y_k^{'}) = \delta^*. \quad (3.3)$$

Such as $\{u_k\}$ and $\{y_k^{'}\}$ are two convergent sequences and θ is continuous as their limits y and $y^{'}$ respectively, we obtain

$$\lim_{k\to\infty} \theta(u_k) = \theta(y), \lim_{k\to\infty} \theta(y_k^{'}) = \theta(y^{'}). \quad (3.4)$$

Using the relations (3.3) and (3.4) we obtain $\theta(y) = \theta(y^{'})$. (3.5)

As $\mathbf{M}$ is a closed map, and using (3.3) and (3.4) we obtain $(y^{'}, \theta(y)) \in M(y, \theta(y))$.

If y did not solve $P(\theta(y))$, $y^{'}$ could improve the value of the objective function of $P(\theta(y))$ at y, i.e. $\Phi(y^{'}) - \theta(y)\Psi(y^{'}) < \Phi(y) - \theta(y)\Psi(y) = 0$, and, using Lemma 2.1, $\theta(y^{'}) < \theta(y)$, this fact contradicts the relation (3.5).

We have proved that y is an optimal solution for $P(\theta(y))$ and Jagannathan$^{'}$s theorem guarantees that y is also an optimal solution for (P).

Conclusions

We have extended Dinkelbach$^{'}$s algorithm developed for the class of convex problems to the general case of fractional programming and have proved its convergences.

We have developed a strategy to speed up the convergence of the extension of Dinkelbach$^{'}$s algorithm. It is based on the truncation of the solution of the subproblems.

The truncation strategy allows the possibility of developing heuristic algorithms for difficult problems of fractional programming by means of the study of the subproblem $P(\delta)$. It is a methodwith which to obtain a decrease sequence for (P) through a decrease sequence for problems of $P(\delta)$.

REFERENCES

[1] Bazaraa, S.H.D. Sherali and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms,* (2nd ed.), John Wiley & Sons Inc., New York, (1993).

[2] Dinkelbach, W., *On Nonlinear Fractional Programming, Management Science,* **13**, 492-498, (1967).

[3] Mond, B., *On Algorithmic Equivalence in Linear Fractional Programming*, Mathematics of Computation, **37**, 185-187, (1981).

[4] Rajendra, V., *On Integer Fractional Linear Programming*, Operations Research Society of India, **30**, 174-176, (1993).

Ovidius University
Bd. Mamaia 124, 900527 Constantza, Romania