Efficient Algorithms and Implementations for Optimizing the Sum of Linear Fractional Functions, with Applications

DANNY Z. CHEN∗
dchen@cse.nd.edu
Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

OVIDIU DAESCU∗,†
daescu@utdallas.edu
Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, USA

YANG DAI‡
yangdai@uic.edu
Department of Bioengineering (M/C 063), University of Illinois at Chicago, Chicago, IL 60607, USA

NAOKI KATOH§ naoki@archi.kyoto-u.ac.jp
Department of Architecture and Architectural Systems, Graduate School of Engineering, Kyoto University Katsura, Nishikyo-ku, Kyoto, Japan

XIAODONG WU∗
xwu@cs.panam.edu
Department of Computer Science, University of Texas-Pan American, 1201 W. University Drive, Edinburg, TX 78541, USA

JINHUI XU∗
jjhui@cse.buffalo.edu
Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260, USA

Received July 10, 2002; Revised January 30, 2003; Accepted May 20, 2003

Abstract. This paper presents an improved algorithm for solving the sum of linear fractional functions (SOLF) problem in 1-D and 2-D. A key subproblem to our solution is the off-line ratio query (OLRQ) problem, which asks to find the optimal values of a sequence of m linear fractional functions (called ratios), each ratio subject to a feasible domain defined by O(n) linear constraints. Based on some geometric properties and the parametric linear programming technique, we develop an algorithm that solves the OLRQ problem in O((m+n) log(m+n)) time. The OLRQ algorithm can be used to speed up every iteration of a known iterative SOLF algorithm, from O(m(m+n)) time to O((m+n) log(m+n)), in 1-D and 2-D. Implementation results of our improved 1-D and 2-D SOLF algorithm have shown that in most cases it outperforms the commonly-used approaches for the SOLF problem. We also apply our techniques to some problems in computational geometry and other areas, improving the previous results.

∗This research was supported in part by the National Science Foundation under Grant CCR-9623585.
†The research of this author was supported in part by National Science Foundation under grant CCF-0430366.
‡Grant-in-Aid of Ministry of Science, Culture and Education of Japan, No. 10780274.
§The research of this author was supported in part by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research on Priority Areas.
Keywords: combinatorial optimization, computational geometry, sum of linear fractional functions, parametric linear programming, robustness

1. Introduction

Recent algorithms for several geometric optimization problems share the following subproblem: find an optimal value for the sum of $m$, 1 or 2-dimensional ($1$-D or $2$-D), linear fractional functions, subject to a set of linear constraints. In this paper, we study the 1-D and 2-D versions of the sum of linear fractionals (SOLF) optimization problem, which is defined as follows:

$$\max_{(x_1, \ldots, x_d) \in S} f(x_1, \ldots, x_d) = \sum_{i=1}^{m} \frac{n_i(x_1, \ldots, x_d)}{d_i(x_1, \ldots, x_d)}$$

such that for each $i = 1, 2, \ldots, m$, $n_i(x_1, \ldots, x_d)$ and $d_i(x_1, \ldots, x_d)$ are linear functions in a fixed $d$-D space $\mathbb{R}^d$, (without loss of generality) $d_i(x_1, \ldots, x_d) > 0$ for any $(x_1, \ldots, x_d) \in S$, the feasible domain $S$ is defined by $n$ linear constraints (i.e., half-spaces in $\mathbb{R}^d$), and $S \neq \emptyset$.

Each linear fractional term $\frac{n_i(x_1, \ldots, x_d)}{d_i(x_1, \ldots, x_d)}$ is called a ratio. The SOLF problem arises in a number of areas, such as geometric optimization, combinatorial optimization and operations research. Quite a few solutions have been given for the SOLF problem (Arkin et al., 1998; Falk and Palocsay, 1992; Konno et al., 1994; Majhi et al., 1997a,b, 1997b; Yamashita, 1997). It is interesting to note that the SOLF related problems were originally studied in economic applications, where the number of variables is usually much bigger than the number of fractional terms. As a result, many previous SOLF algorithms were designed to target problems with only a few fractional terms (less than 10) for a reasonable running time (Falk and Palocsay, 1992; Konno et al., 1994; Yamashita, 1997). On the other hand, there are some general-purpose heuristic packages which can generate local optimal solutions (Ingber, http://www.ingber.com/#ASA-CODE; Jelasity and Dombi, ftp://ftp.jate.u-szeged.hu/pub/math/optimization/GAS/). However, they usually run in a long time yet obtain solutions without quality guarantee.

Some complexity results on fractional functions and related problems are known. For example, solving $n$-variable fractional programs with sum of ratios is NP-complete, even when the sum has only two or few terms (Freund and Jarre, 2001). Chandrasekaran (1977) has considered the minimum ratio spanning tree (MRST) problem: given a graph $G$ where every edge $e$ has associated two values (cost and weight), $a_e$ and $b_e$, find a spanning tree $T$ of $G$ such that the ratio $\frac{\sum_{e \in T} a_e}{\sum_{e \in T} b_e}$ is minimized. He proved that the problem is NP-complete when the denominator of the MRST objective function is allowed to take on negative values. In contrast, there are known polynomial time solutions for some ratio optimization problems. For example, Barros (1998) has shown that a number of facility location problems with a single fractional function can be solved in polynomial time. The MRST problem with positive costs and weights can be solved in polynomial time (Chandrasekaran, 1977). More general, Meggido (1979) showed that if a combinatorial optimization problem with a linear cost model admits a polynomial time solution, then the minimum cost to weight ratio problem can be also solved in polynomial time. Skiscim and Palocsay (2001)