Journal of Computational and Applied Mathematics 234 (2010) 733–742

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Solving the sum-of-ratios problems by a harmony search algorithm

Majid Jaberipour *, Esmaile Khorram

Department of Applied Mathematics, Faculty of Mathematics and Computer Science, Amirkabir University of Technology, No. 424, Hafez Ave., 15914 Tehran, Iran

ARTICLE INFO

Article history:
Received 24 July 2009
Received in revised form 13 January 2010

Keywords:
Harmony search
Fractional programming
Sum-of-ratios
Meta-heuristic

ABSTRACT

The sum-of-ratios problems have numerous applications in economy and engineering. The sum-of-ratios problems are considered to be difficult, as these functions are highly nonconvex and multimodal. In this study, we propose a harmony search algorithm for solving a sum-of-ratios problem. Numerical examples are also presented to demonstrate the effectiveness and robustness of the proposed method. In all cases, the solutions obtained using this method are superior to those obtained from other methods.

1. Introduction

In this paper, we consider a special class of fractional programming problem in the following form:

$$\begin{aligned} \min/\max \quad & \sum_{i=1}^{p} \frac{f_i(\mathbf{x})}{g_i(\mathbf{x})} \\ & h_k(\mathbf{x}) \geq 0, \quad k = 1, \ldots, K \\ & m_j(\mathbf{x}) = 0, \quad j = 1, \ldots, J \\ & x_i^l \leq x_i \leq x_i^u \quad i = 1, \ldots, n \end{aligned} \tag{1}$$

where f, g, h and m are linear, quadratic, or more general functions. The sum-of-ratios problems have attracted the interest of researchers since at least the 1970s. During the past 10 years, interest in these problems has been especially strong. In part, this is because, from a practical point of view, these problems have spawned a wide variety of important applications, especially in government contracting, transportation science, finance, economics, engineering, etc. In addition, from a research point of view, these problems pose significant theoretical and computational challenges. This is mainly due to the fact that these problems are global optimization problems, i.e. they are known to generally possess multiple local optima that are not global optima. Fractional problems have been studied since the 1960s when Charnes and Cooper [1] proposed their famous transformation for rewriting the problem of maximizing a ratio of linear functions over linear constraints as an ordinary linear program. Several algorithms have been proposed for solving nonlinear fractional programming problems [2–8], but in the most considered problems the feasible regions are polyhedrons or convex sets. Other contributions to the field include an approach exploiting monotonicity in the objective by Phuong and Tuy [9] which can also handle products of ratios and max–min problems, and Dür et al. [10] presented a unified method based on the Multistart Pure Adaptive Search. Shen et al. [11,12] solved the generalized fractional programming problem which contains various variants such as a sum or product of a finite number of ratios of linear functions, polynomial fractional programming, generalized geometric programming using the branch and bound method. Related work on fractional programming can be referred to [13–15]. The

* Corresponding author. Tel.: +98 21 64542549; fax: +98 21 66497930.
E-mail addresses: Majid.Jaberipour@gmail.com (M. Jaberipour), eskhor@aut.ac.ir (E. Khorram).

0377-0427/$ – see front matter
doi:10.1016/j.cam.2010.01.013

aim of the present paper is to develop a unified approach to many different types of the sum-of-ratios problems. In response to this demand, in this study, a harmony search algorithm has been presented. Geem et al. [16] introduced basic harmony search (HS) that draws its inspiration not from a biological or physical process like most other meta-heuristic optimization techniques, but from an artistic one—the improvisation process of musicians seeking a wonderful harmony. Geem et al. [16] explained the analogy between optimization and musical performance that embodies the spirit and language of HS. The effort to find the harmony in music is analogous to find the optimality in an optimization process and the musicians improvisations are analogous to local and global search schemes in optimization techniques. HS methods have been applied to a diverse range of problems from structural design to solving Sudoku puzzles, from musical composition to medical imaging, from heat exchanger design to course timetabling.

The paper is organized as follows. In Section 2, we describe basic harmony search and a proposed harmony search algorithm. In Section 3, an illustrative example is presented to compare a proposed harmony search algorithm with the basic harmony search. Additionally, the effect of HMCR, PAR and bw is investigated on a proposed harmony search. In Section 4, some standard benchmark examples are also presented to demonstrate the effectiveness and robustness of the presented approach. Finally, conclusion is indicated in Section 5.

2. Harmony search meta-heuristic algorithm

The HS algorithm conceptualizes a behavioral phenomenon of musicians in the improvisation process, where each musician continues to experiment and improve his or her contribution in order to search for a better state of harmony [17,18]. This section describes the HS algorithm based on the heuristic algorithm that searches for a globally optimized solution. The procedure for a harmony search, which consists of steps 1–5.

Step 1. Initialize the optimization problem and algorithm parameters.

Step 2. Initialize the harmony memory (HM).

Step 3. Improvise a new harmony from the HM.

Step 4. Update the HM.

Step 5. Repeat Steps 3 and 4 until the termination criterion is satisfied.

Step 1. Initialize the optimization problem and algorithm parameters. First, the optimization problem is specified as follows:

$$\text{Minimize } f(X) \quad \text{subject to } x_i \in X_i = 1, 2, \ldots, N$$

where $f(X)$ is an objective function; X is the set of decision variables; N is the number of decision variables; X_i is the set of the possible range of values for each decision variable, that is, $x_i^L \le x_i \le x_i^U$; and x_i^L and x_i^U are the lower and upper bounds for each decision variable, respectively. The algorithm requires several parameters: Harmony Memory Size (HMS), Maximum number of Improvisations (MaxImp), Harmony Memory Considering Rate (HMCR), Pitch Adjusting Rate (PAR), Bandwidth vector used in (bw).

In *Step* 2, the HM matrix is initially filled with as many randomly generated solution vectors as the HMS, as well as with the corresponding function values of each random vector, $f(X)$.

$$\text{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \ldots & x_N^1 & | & f(\mathbf{X}^1) \\ x_1^2 & x_2^2 & \ldots & x_N^2 & | & f(\mathbf{X}^2) \\ \vdots & \vdots & \ddots & \vdots & | & \vdots \\ x_1^{HMS} & x_2^{HMS} & \ldots & x_N^{HMS} & | & f(\mathbf{X}^{HMS}) \end{bmatrix}.$$

Step 3. A new harmony vector, $\acute{X} = (\acute{x}_1, \acute{x}_2, \ldots, \acute{x}_N)$, is improvised based on the following three mechanisms: (1) random selection, (2) memory consideration, and (3) pitch adjustment. In the random selection, the value of each decision variable, in the new harmony vector is randomly chosen within the value range with a probability of (1-HMCR). The HMCR, which varies between 0 and 1, is the rate of choosing one value from the historical values stored in the HM, and (1-HMCR) is the rate of randomly selecting one value from the possible range of values.

$$\acute{x}_i \leftarrow \begin{cases} \acute{x}_i \in \{x_i^1, x_i^2, \ldots, x_i^{\text{HMS}}\} & \text{with probability HMCR,} \\ \acute{x}_i \in X_i & \text{with probability (1-HMCR).} \end{cases}$$

The value of each decision variable obtained by the memory consideration is examined to determine whether it should be pitch-adjusted. This operation uses the PAR parameter, which is the rate of pitch adjustment as it should be pitch-adjusted to neighboring pitches with a probability of HMCR $\times$ PAR, while the original pitch obtained in the memory consideration is kept with a probability of HMCR $\times$ (1-PAR). If the pitch adjustment decision for $\acute{x}_i$ is made with a probability of PAR, $\acute{x}_i$ is replaced with $\acute{x}_i \pm u(-1, 1) \times \text{bw}$, where bw is an arbitrary distance bandwidth for the continuous design variable, and $u(-1, 1)$ is a uniform distribution between -1 and 1. The value of (1-PAR) sets the rate of performing nothing. Thus, pitch adjustment is applied to each variable as follows:

$$\acute{x}_i \leftarrow \begin{cases} \acute{x}_i \pm u(-1, 1) \times \text{bw} & \text{with probability HMCR } \times \text{ PAR} \\ \acute{x}_i & \text{with probability HMCR } \times \text{ (1-PAR).} \end{cases}$$