International Journal of Approximate Reasoning 82 (2017) 1–21

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Revisiting Karnik–Mendel Algorithms in the framework of Linear Fractional Programming

Tufan Kumbasar

Control and Automation Engineering Department, Faculty of Electrical and Electronics Engineering, Istanbul Technical University, Maslak, TR-34469, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 6 April 2016
Received in revised form 19 September 2016
Accepted 28 November 2016
Available online 2 December 2016

Keywords:
Type-2 fuzzy sets and systems
Type reduction
Centroid calculation
Karnik–Mendel Algorithms
Linear Fractional Programming
Dinkelbach's Algorithm

ABSTRACT

For Interval Type-2 (IT2) fuzzy sets and systems, calculating the centroid and performing type reduction are operations that must be taken into consideration. The Karnik–Mendel Algorithm (KMA) and its enhancements have been usually employed to perform these operations. In KMAs, these IT2 fuzzy operations are defined as nonlinear optimization problems which are solved iteratively by finding the optimal Switching Points (SPs). In this study, we will examine these optimization problems and will reveal the nature of KMAs in the framework of Linear Fractional Programming (LFP). In this context, we will firstly transform IT2 fuzzy operations into 0-1 LFP problems and then we will show that there exists a direct relationship between the SPs of the KMAs and the solution vectors of the defined LFP problems. Hence, the meaning of the SPs will be revealed in the framework of LFP theory. Thus, it will be concluded that the KMAs can be seen as LFP methods. Then, by taking account of the LFP representations, we will show that it is possible to perform the IT2 fuzzy operations via well-known LFP methods, namely the Charnes Cooper Transformation and the Dinkelbach's Algorithm (DA). It is believed that the presented LFP approaches will be very helpful in employing IT2 fuzzy sets and systems in different programming languages since they only use and employ built-in Linear Programming (LP) routines. However, since LP solvers are used, it will be concluded that the LFP based approaches are computationally intensive and thus might not be feasible for real-time applications. Therefore, we will present a computationally efficient implementation of presented the DA based method, namely the KMK Algorithm (KMKA). Then, through analytical investigations on the solution vectors and optimality conditions of the KMKA, it will be revealed that the KMA is algorithmic equivalent to the presented implementation of the DA based KMKA. Thus, it will be revealed that the commonly employed KMA and its enhancements can be seen as efficient implementations of the DA in the framework of LFP. Based on this connection, since the DA is an adaptation of the Newton method to a non-differentiable context, it will be also concluded that the discrete version of KMA can be also seen equivalent to a root-finding problem. We will present comparative numerical results to validate the analytical derivations and observations.

E-mail address: kumbasart@itu.edu.tr.

http://dx.doi.org/10.1016/j.ijar.2016.11.019

1. Introduction

Computing the centroid of Interval Type-2 (IT2) Fuzzy Sets (FSs) and performing Type-Reduction (TR) for IT2 Fuzzy Logic Systems (FLSs) are the most important operations which have been developed for such FSs and FLSs [1–8]. These operations are defined as follows [9]:

For given x_n and w_n:

$$x_n \in X_n \equiv [\underline{x}_n, \bar{x}_n] \quad \text{where } \underline{x}_n \leq \bar{x}_n,\ n = 1, 2, \ldots N \tag{1}$$

$$w_n \in W_n \equiv [\underline{w}_n, \overline{w}_n] \quad \text{where } \underline{w}_n \leq \overline{w}_n,\ n = 1, 2, \ldots N \tag{2}$$

Compute

$$Y = \frac{\sum_{n=1}^{N} X_n W_n}{\sum_{n=1}^{N} W_n} \equiv [\underline{y}, \bar{y}] \tag{3}$$

where $\underline{y}$ and $\bar{y}$ are the extrema values of the problems:

$$\text{P1:} \quad \underline{y} = \min_{\substack{\forall x_n \in [\underline{x}_n, \bar{x}_n] \\ \forall w_n \in [\underline{w}_n, \overline{w}_n]}} \frac{\sum_{n=1}^{N} x_n w_n}{\sum_{n=1}^{N} w_n} \tag{4}$$

$$\text{P2:} \quad \bar{y} = \max_{\substack{\forall x_n \in [\underline{x}_n, \bar{x}_n] \\ \forall w_n \in [\underline{w}_n, \overline{w}_n]}} \frac{\sum_{n=1}^{N} x_n w_n}{\sum_{n=1}^{N} w_n} \tag{5}$$

In computing the centroid of an IT2-FSs ($\tilde{A}$) [1–5], x_n ($\underline{x}_n = \bar{x}_n$) represent discretizations of the primary variable x, $[\underline{w}_n, \overline{w}_n]$ corresponds to the interval membership grade of x_n and Y is the resulting centroid of the $\tilde{A}$. Here, the membership grade is defined with a Lower Membership Function (LMF) $\underline{w}_n$ and an Upper Membership Function (UMF) $\overline{w}_n$ which create the Footprint of Uncertainty (FOU) of IT2-FS [2]. In the center of sets TR of IT2-FLSs, Y is the type-reduced set, X_n represents the centroid of the consequent IT2-FS of the nth rule and W_n is the firing level of the corresponding rule [9–14]. A same problem is also defined for computing the fuzzy weighted average [9,15,16].

The presented nonlinear optimization problems P1 and P2 have been usually solved by employing the Karnik–Mendel Algorithms (KMAs). The KMAs solve these problems by finding iteratively the optimal combination of the variables $\underline{w}_n$ and $\overline{w}_n$, i.e. the Switching Points (SPs) L and R [17]. Thus, the original KMA has been seen as computationally intensive, especially when the dimension N is large [9,11]. Several studies have been presented to understand the nature of the KMAs [4–8,17–19] which leaded to more efficient algorithms [9] such as the Enhanced KMA (EKMA) [17]. The KMA and its enhancements have been firstly proven to be superexponentially convergent [19] and then quadratically convergent through further investigations [5]. Moreover, it has been shown in [5] that the continuous version of the KMA is equivalent to root-finding problem which can be solved with Newton–Raphson method. Recently, it has been shown that IT2 fuzzy operations can be also handled as Linear Fractional Programming (LFP) problems [20].

In this study, we will examine the nonlinear optimization problems P1 and P2 in the well-developed LFP theory and will reveal the nature of the KMAs in the framework of LFP. It will be firstly shown that the defined P1 and P2 optimization problems can be reformulated as 0-1 LFP problems. Then, it will be shown that there exists a direct relationship between the SPs of the KMAs and the solution vectors of the defined LFP problems. In other words, the meaning of the SPs will be revealed in the framework of LFP theory. We will then take advantage of 0-1 LFP representations to solve the optimization problems P1 and P2 via the widely employed LFP methods which are the Charnes Cooper Transformation (CCT) [21] and the Dinkelbach's Algorithm (DA) [22]. We will demonstrate that the CCT and DA based approaches can be employed in a straightforward manner since both LFP methods use and employ basic built-in Linear Programming (LP) routines. However, we will also show that these LFP based methods are computationally intensive in comparison to the KMA since they use LP solvers which are time consuming. Motivated by this drawback, we will present a computationally efficient implementation of the DA based approach, namely the KMK Algorithm (KMKA). Moreover, through analytical investigations, it will be shown that there exists a direct relation of the efficient implementation of the DA (the KMKA) and the KMA by connecting their solution vectors and their optimality conditions. Then, in the light of these derived connections, it will be concluded that KMKA and KMA are algorithmic equivalent. Thus, it will be concluded that the commonly employed KMA and its enhancements can be seen as efficient implementations of the DA in the framework of LFP. Based on this connection, since the DA is an adaptation of the Newton method to a non-differentiable context [23], it will be also concluded that the discrete version of KMA can be also seen equivalent to a root-finding problem. Although it is not the goal of the paper to present an efficient TR/CC algorithm, we will present numerical results to show that the KMKA is a relatively better implementation of DA in comparison with the KMA and EKMA.

Section 2 gives preliminaries about the KMAs and LFP. Section 3 presents the transformation of the problems P1 and P2 into LFP problems and the connection of presented LFP problems to the KMAs. Section 4 gives the DA and CCT based TR/CC approaches. Section 5 introduces the computationally efficient KMKA and its connection to the KMA. Section 6 presents conclusions.