ELSEVIER

Fuzzy Sets and Systems 87 (1997) 39–45

FUZZY sets and systems

An efficient algorithm for fuzzy weighted average

Dong Hoon Lee[1], Daihee Park*

Department of Computer Science, Korea University, 208 Seochang-Dong, Chochiwon, Chung-nam 339-700, South Korea

Received January 1995; revised January 1996

Abstract

In multisensor intelligent systems, the information fusion plays an important role. Several algorithms have been proposed for the purpose of aggregating imprecise sensory information represented by fuzzy numbers. This paper proposes an efficient algorithm to compute fuzzy weighted average, which turned out to be superior to the previous works by reducing the number of comparisons and arithmetic operations to O($n \log n$). © 1997 Elsevier Science B.V.

Keywords: Fuzzy number; Fuzzy arithmetic; Multiple criteria evaluation; Fuzzy weighted average; Computational efficiency

1. Introduction

Information fusion is used whenever several sensors are employed in a system, in order to reduce uncertainty and resolve the ambiguity often present in the information from a single sensor. Consequently, the addition of extra sensors, providing redundant and complementary information, offers the capability of resolving most complex situations and leads to a richer description of the world.

Several approaches for information fusion have been proposed in the literature. Among them are probability theory, Dempster–Shafer theory, neural networks and fuzzy set theory [2]. However, since the fusion of information is often made more difficult by problems of uncertainty characterized by vagueness, inexactness and ill-definedness, it is more appropriate to employ the fuzzy set theory in the information fusion systems.

We shall assume that the reader is familiar with the basics of the fuzzy set theory. The following definitions relating to fuzzy weighted average are mostly from [4]. To generalize the fuzzy weighted average, let $A_1, A_2, \ldots, A_n$, and $W_1, W_2, \ldots, W_n$ be the fuzzy numbers defined on the universes $X_1, X_2, \ldots, X_n$, and $Z_1, Z_2, \ldots, Z_n$, respectively. If f is a function which maps from $X_1 \times X_2 \times \cdots \times X_n \times Z_1 \times Z_2 \times \cdots \times Z_n$ to the universe Y, then the *fuzzy weighted average* y is defined as follows:

$$y = f(x_1, x_2, \ldots, x_n, w_1, w_2, \ldots, w_n) = \frac{x_1 w_1 + x_2 w_2 + \cdots + x_n w_n}{w_1 + w_2 + \cdots + w_n},$$

where for each $i = 1, 2, \ldots, n$, $x_i \in X_i$, $w_i \in Z_i$, and $w_1 + w_2 + \cdots + w_n > 0$. Let μ_B be the membership

* Corresponding author. Supported in part by KOSEF (#951-0906-093-z).

[1] Supported in part by KOSEF (#951-0903-047-z).

PII S0165-0114(96)00027-9

function of the fuzzy image B of $A_1, A_2, \ldots, A_n$, $W_1, W_2, \ldots, W_n$ through f. Then by the extension principle,

$$\mu_B(y) = \max_{\substack{x_i \in X_i, w_i \in Z \\ i=1,2,\ldots,n \\ y=f(x_1,x_2,\ldots,x_n)}} \{\min[\mu_{A_1}(x_1), \ldots, \mu_{A_n}(x_n), \mu_{W_1}(w_1), \ldots, \mu_{W_n}(w_n)]\},$$

where for each $i = 1, 2, \ldots, n$, μ_{A_i} and μ_{W_i} are the membership functions of fuzzy number A_i and W_i, respectively.

Dong and Wong [3] firstly proposed an algorithm to compute the fuzzy weighted average (FWA) based on the α-cut representation of fuzzy sets and combinatorial interval analysis. However, since this algorithm requires an exponential computation, it becomes very complicated and cumbersome as the number of fused information increases. Subsequently, Liou and Wang [4] suggested an improved fuzzy weighted average algorithm (IFWA) to simplify the computational process with modifications on steps (3) and (4) of the FWA. The IFWA algorithm improved the FWA algorithm in terms of computation time, i.e., it requires $O(n^2)$ comparisons and arithmetic operations.

In this paper, we propose an efficient fuzzy weighted average algorithm (EFWA) which turned out to be superior to the previous works by reducing the number of comparisons and arithmetic operations to $O(n \log n)$.

This paper is organized as follows. In Section 2, the theoretical background for the EFWA algorithm is constructed and verified. In addition, an efficient algorithm based on the developed theory is introduced. In Section 3, a numerical example is given for the sake of illustration. Finally, Section 4 concludes with discussions and a brief summary.

2. The efficient fuzzy weighted average algorithm

2.1. Theoretical backgrounds

Suppose that for each $i = 1, 2, \ldots, n$, x_i and w_i have the corresponding interval $[a_i, b_i]$ and $[c_i, d_i]$ with $c_i \geqslant 0$, respectively. Define $f_L(w_1, w_2, \ldots, w_n) = f(a_1, a_2, \ldots, a_n, w_1, w_2, \ldots, w_n)$ and $f_U(w_1, w_2, \ldots, w_n) = f(b_1, b_2, \ldots, b_n, w_1, w_2, \ldots, w_n)$. Liou and Wang [4] have shown the following result.

Proposition 1 (Liou and Wang [4]). *$\min f(x_1, x_2, \ldots, x_n, w_1, w_2, \ldots, w_n) = \min f_L(w_1, w_2, \ldots, w_n)$ and $\max f(x_1, x_2, \ldots, x_n, w_1, w_2, \ldots, w_n) = \max f_U(w_1, w_2, \ldots, w_n)$.*

Let $S = (e_1, e_2, \ldots, e_n)$ be an n-tuple in $Z_1 \times Z_2 \times \cdots \times Z_n$. For each $i = 1, 2, \ldots, n$, let

$$\delta_{S_i} = \frac{(a_1 - a_i)e_1 + (a_2 - a_i)e_2 + \cdots + (a_n - a_i)e_n}{e_1 + e_2 + \cdots + e_n},$$

$$\zeta_{S_i} = \frac{(b_1 - b_i)e_1 + (b_2 - b_i)e_2 + \cdots + (b_n - b_i)e_n}{e_1 + e_2 + \cdots + e_n}.$$

Let $\delta_{S_0} = \delta_{S_1} + 1$ and $\zeta_{S_0} = \zeta_{S_1} + 1$. Note that $f_L(e_1, e_2, \ldots, e_n) = a_1 + \delta_{S_1} = a_2 + \delta_{S_2} = \cdots = a_n + \delta_{S_n}$ and $f_U(e_1, e_2, \ldots, e_n) = b_1 + \zeta_{S_1} = b_2 + \zeta_{S_2} = \cdots = b_n + \zeta_{S_n}$. Throughout the paper, we assume that an n-tuple is in $Z_1 \times Z_2 \times \cdots \times Z_n$, unless stated otherwise.

Lemma 2. *For each n-tuple S,* (i) *if $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$, there exists only one integer i, $0 \leqslant i < n$, such that $\delta_{S_i} > 0$ and $\delta_{S_{i+1}} \leqslant 0$, and* (ii) *if $b_1 \leqslant b_2 \leqslant \cdots \leqslant b_n$, there exists only one integer j, $0 \leqslant j < n$, such that $\zeta_{S_i} > 0$ and $\zeta_{S_{i+1}} \leqslant 0$.*

Proof. We shall show (i) only because (ii) can be shown similarly.

Let S be an n-tuple and assume that a's are sorted in nondecreasing order. Since $a_1 + \delta_{S_1} = a_2 + \delta_{S_2} = \cdots = a_n + \delta_{S_n}$, $\delta_{S_0} (= \delta_{S_1} + 1) > \delta_{S_1} \geqslant \delta_{S_2} \geqslant \cdots \geqslant \delta_{S_n}$. By definition of δ's, $\delta_{S_0} > 0$ and $\delta_{S_n} \leqslant 0$, and hence there should be an integer i, $0 \leqslant i < n$, such that $\delta_{S_i} > 0$ and $\delta_{S_{i+1}} \leqslant 0$. It is clear that such i is unique. □

Let S be an n-tuple. Suppose a's are sorted in nondecreasing order. The δ-threshold of S is an integer i such that $\delta_{S_i} > 0$ and $\delta_{S_{i+1}} \leqslant 0$. If b's are sorted in nondecreasing order, the ζ-threshold of S is an integer i such that $\zeta_{S_i} > 0$ and $\zeta_{S_{i+1}} \leqslant 0$.

Lemma 3. (i) *Let $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$. Then there exists one and only one $i \in \{0, 1, 2, \ldots, n-1\}$ such* *that the* δ*-threshold of* $S = (d_1, \ldots, d_i, c_{i+1}, \ldots, c_n)$ *is* i. (ii) *Let* $b_1 \leqslant b_2 \leqslant \cdots \leqslant b_n$. *Then there exists one and only one* $i \in \{0, 1, 2, \ldots, n-1\}$ *such that the* ζ*-threshold of* $S = (c_1, \ldots, c_i, d_{i+1}, \ldots, d_n)$ *is* i.

Proof. We prove (i) only because same reasoning holds for the proof of (ii). We shall show the existence first. For each $i = 0, 1, \ldots, n$, let S^i denote the n-tuple of the form $(d_1, \ldots, d_i, c_{i+1}, \ldots, c_n)$. Suppose on the contrary that $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$ and there is no n-tuple S^i with the δ-threshold i. For each $i = 0, 1, \ldots, n-1$, $\delta_{S^i_{i+1}} > 0$ if the δ-threshold of S^i is greater than i and $\delta_{S^i_i} \leqslant 0$ if it is less than i. If $\delta_{S^i_{i+1}} > 0$ then $\delta_{S^{i+1}_{i+1}} > 0$. (We note that the numerator of $\delta_{S^{i+1}_{i+1}}$ equals that of $\delta_{S^i_{i+1}}$ which is greater than 0.) Furthermore, $\delta_{S^{i+1}_{i+2}} > 0$ because the δ-threshold of S^{i+1} cannot be $i+1$. This inductively implies that $\delta_{S^{n-1}_n} > 0$. But $\delta_{S^{n-1}_n} \leqslant 0$, a contradiction. If $\delta_{S^i_i} \leqslant 0$, then $\delta_{S^{i-1}_i} \leqslant 0$ and $\delta_{S^{i-1}_{i-1}} \leqslant 0$ by the similar reasoning stated above. This means that $\delta_{S^0_1} \leqslant 0$. The numerator of $\delta_{S^0_1}$ is $(a_1 - a_1)c_1 + (a_2 - a_1)c_2 + \cdots + (a_n - a_1)c_n$ and cannot be less than 0 and $\delta_{S^0_1}$ should be 0. By definition of δ, $\delta_{S^0_0} = \delta_{S^0_1} + 1 = 1$. It follows that the δ-threshold of S^0 is 0. This is again a contradiction, which proves the existence.

Next, we show the uniqueness.

Suppose not. Suppose there are two n-tuples S and T such that $S = (d_1, \ldots, d_i, c_{i+1}, \ldots, c_n)$ and $T = (d_1, \ldots, d_j, c_{j+1}, \ldots, c_n)$ have δ-thresholds i and j, respectively, and $i \neq j$. Without loss of generality, assume that $i < j$. Let $E_s = (a_1 - a_j)d_1 + \cdots + (a_i - a_j)d_i + (a_{i+1} - a_j)c_{i+1} + \cdots + (a_n - a_j)c_n$ and $E_T = (a_1 - a_j)d_1 + \cdots + (a_j - a_j)d_j + (a_{j+1} - a_j)c_{j+1} + \cdots + (a_n - a_j)c_n$. Then $\delta_{S_j} = E_S/(d_1 + \cdots + d_i + c_{i+1} + \cdots + c_n)$ and $\delta_{T_j} = E_T/(d_1 + \cdots + d_j + c_{j+1} + \cdots + c_n)$.

Since $d_{i+1} \geqslant c_{i+1}, d_{i+2} \geqslant c_{i+2}, \ldots, d_j \geqslant c_j$, it is obvious that $E_S \geqslant E_T$. By definition of the δ-threshold, $\delta_{T_j} > 0$ and hence $E_S \geqslant E_T > 0$. Now, we have $\delta_{S_j} > 0$. This contradicts the supposition that the δ-threshold of S is i $(< j)$ and so $\delta_{S_j} \leqslant 0$. This proves the result. □

Let $S = (e_1, e_2, \ldots, e_n)$ be an n-tuple. For β_i in Z_i, define $f_L(e_1, e_2, \ldots, e_n \mid \beta_i) = f_L(e_1, \ldots, e_{i-1}, \beta_i, e_{i+1}, \ldots, e_n)$ and $f_U(e_1, e_2, \ldots, e_n \mid \beta_i) = f_U(e_1, \ldots, e_{i-1}, \beta_i, e_{i+1}, \ldots, e_n)$.

The following theorem shows that the problems to find the values of $\min f_L(w_1, w_2, \ldots, w_n)$ and $\max f_U(w_1, w_2, \ldots, w_n)$ can be solved simply by (i) sorting a's and b's in nondecreasing order and (ii) finding n-tuples $(d_1, \ldots, d_i, c_{i+1}, \ldots, c_n)$ and $(c_1, \ldots, c_j, d_{j+1}, \ldots, d_n)$ with the δ-threshold i and the ζ-threshold j, respectively.

Theorem 4. (i) *Let* $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$. *If an* n*-tuple* $(d_1, \ldots, d_i, c_{i+1}, \ldots, c_n)$ *has the* δ*-threshold* i, $0 \leqslant i < n$, *then* $f_L(d_1, \ldots, d_i, c_{i+1}, \ldots, c_n) = \min f_L(w_1, w_2, \ldots, w_n)$. (ii) *Let* $b_1 \leqslant b_2 \leqslant \cdots \leqslant b_n$. *If an* n*-tuple* $(c_1, \ldots, c_j, d_{j+1}, \ldots, d_n)$ *has the* ζ*-threshold* j, $0 \leqslant j < n$, *then* $f_U(c_1, \ldots, c_j, d_{j+1}, \ldots, d_n) = \max f_U(w_1, w_2, \ldots, w_n)$.

Proof. We show (i) first. Suppose to the contrary that $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$ and there exists an n-tuple $S = (d_1, \ldots, d_i, c_{i+1}, \ldots, c_n)$ with the δ-threshold i such that $f_L(d_1, \ldots, d_i, c_{i+1}, \ldots, c_n)$ is greater than $\min f_L(w_1, w_2, \ldots, w_n)$. Thus, there exists an n-tuple $T = (e_1, e_2, \ldots, e_n)$ with $T \neq S$ such that $f_L(e_1, e_2, \ldots, e_n) = \min f_L(w_1, w_2, \ldots, w_n)$. Let k be the δ-threshold of the n-tuple T. By Lemma 3, $(e_1, e_2, \ldots, e_n) \neq (d_1, d_2, \ldots, d_k, c_{k+1}, \ldots, c_n)$. So, we have two cases: there is an $e_j (j \leqslant k)$ with $e_j \neq d_j$ or there is an $e_j (j > k)$ with $e_j \neq c_j$.

Case 1 (There is an $e_j (j \leqslant k)$ with $e_j \neq d_j$): Let j be the smallest integer such that $e_j \neq d_j$ (i.e., $e_j < d_j$). Let $E = (a_1 - a_j)e_1 + (a_2 - a_j)e_2 + \cdots + (a_n - a_j)e_n$. Since the δ-threshold of T is $k (\geqslant j)$, $\delta_{T_j} > 0$ and hence $E > 0$. We observe that $\delta_{T_j} = E/(e_1 + e_2 + \cdots + e_n) > E/(e_1 + \cdots + e_{j-1} + d_j + e_{j+1} + \cdots + e_n)$. So, $f_L(e_1, e_2, \ldots, e_n) = a_j + \delta_{T_j} > a_j + E/(e_1 + \cdots + e_{j-1} + d_j + e_{j+1} + \cdots + e_n) = f_L(e_1, e_2, \ldots, e_n \mid d_j)$. This is a contradiction to the fact that $f_L(e_1, e_2, \ldots, e_n) = \min f_L(w_1, w_2, \ldots, w_n)$.

Case 2 (There is an $e_j (j > k)$ with $e_j \neq c_j$): In this case, we assume that $e_1 = d_1, e_2 = d_2, \ldots, e_k = d_k$ by using the result in the Case 1. Let j be the smallest integer such that $e_j \neq c_j$ (i.e., $e_j > c_j$) and $j > k$. Let $E = (a_1 - a_j)e_1 + (a_2 - a_j)e_2 + \cdots + (a_n - a_j)e_n$. Since the δ-threshold of T is k, $\delta_{T_j} \leqslant 0$ and hence $E \leqslant 0$. If $E < 0$, then $\delta_{T_j} = E/(e_1 + e_2 + \cdots + e_n) > E/(e_1 + \cdots + e_{j-1} + c_j + e_{j+1} + \cdots + e_n)$. This implies that $f_L(e_1, e_2, \ldots, e_n) > f_L(e_1, e_2, \ldots, e_n \mid c_j)$, which contradicts the supposition that $f_L(e_1, e_2, \ldots, e_n) = \min f_L(w_1, w_2, \ldots, w_n)$. So,

E must be zero. If $e_l = c_l$ for each $l = j+1, \ldots, n$ or $a_l - a_j = 0$ for every l with $e_l \neq c_l$ and $j < l \leqslant n$, the value of E is equal to the value of $(a_1 - a_j)d_1 + \cdots + (a_k - a_j)d_k + (a_{k+1} - a_j)c_{k+1} + \cdots + (a_n - a_j)c_n$ and thus $f_L(e_1, e_2, \ldots, e_n) = f_L(d_1, \ldots, d_k, c_{k+1}, \ldots, c_n) = \min f_L(w_1, w_2, \ldots, w_n)$. Let U be the n-tuple $(d_1, \ldots, d_k, c_{k+1}, \ldots, c_n)$. Since $a_1 + \delta_{T_1} = a_2 + \delta_{T_2} = \cdots = a_n + \delta_{T_n} = f_L(e_1, e_2, \ldots, e_n)$ and $a_1 + \delta_{U_1} = a_2 + \delta_{U_2} = \cdots = a_n + \delta_{U_n} = f_L(d_1, \ldots, d_k, c_{k+1}, \ldots, c_n)$, $\delta_{T_i} = \delta_{U_i}$ for each $i = 1, \ldots, n$. Hence, the δ-threshold of U should be the δ-threshold of T which is k. By Lemma 3, $S = U$. But we have supposed that $f_L(d_1, \ldots, d_i, c_{i+1}, \ldots, c_n) > \min f_L(w_1, w_2, \ldots, w_n)$. This is impossible and there should exist e_l, $j < l \leqslant n$, such that $e_l > c_l$ and $a_l - a_j > 0$. Let $E' = (a_1 - a_j)e_1 + \cdots + (a_{l-1} - a_j)e_{l-1} + (a_l - a_j)c_l + (a_{l+1} - a_j)e_{l+1} + \cdots + (a_n - a_j)e_n$. Since $E(=0) > E'$,

$$a_j + \frac{E}{e_1 + e_2 + \cdots + e_n} > a_j + \frac{E'}{e_1 + \cdots + e_{l-1} + c_l + e_{l+1} + \cdots + e_n}.$$

So $f_L(e_1, e_2, \ldots, e_n) > f_L(e_1, e_2, \ldots, e_n \mid c_l)$. Again this is a contradiction.

In both cases, we have contradictions. This proves (i). The proof of (ii) is similar to that of (i), hence omitted here. □

2.2. EFWA algorithm

We now describe the efficient FWA (EFWA, for short) to find $L = \min f_L(w_1, w_2, \ldots, w_n)$ and $U = \max f_U(w_1, w_2, \ldots, w_n)$ by modifying step (3) and step (4) of the FWA algorithm [3]. The basic idea of the algorithm is to sort a's and b's in nondecreasing order and find two n-tuples $(d_1, \ldots, d_i, c_{i+1}, \ldots, c_n)$ with the δ-threshold i and $(c_1, \ldots, c_j, d_{j+1}, \ldots, d_n)$ with the ζ-threshold j, which are the n-tuples such that $f_L(d_1, \ldots, d_i, c_{i+1}, \ldots, c_n) = \min f_L(w_1, w_2, \ldots, w_n)$ and $f_U(c_1, \ldots, c_j, d_{j+1}, \ldots, d_n) = \max f_U(w_1, w_2, \ldots, w_n)$ by Theorem 4. In the following, we briefly describe how to find the n-tuple $(d_1, \ldots, d_i, c_{i+1}, \ldots, c_n)$ with the δ-threshold i. The n-tuple $(c_1, \ldots, c_j, d_{j+1}, \ldots, d_n)$ with the ζ-threshold j can be found similarly. Let S^i be of the form $(d_1, \ldots, d_i, c_{i+1}, \ldots, c_n)$. As already shown in the proof of Lemma 3, if the δ-threshold of S^i is not i and $\delta_{S^i_{i+1}} > 0$ ($\delta_{S^i_i} \leqslant 0$), then the tuple we are searching is S^j for some $j > i$ ($j < i$, resp.). So, to find S^j with the δ-threshold j, the algorithm uses the well-known binary search method [1].

Algorithm EFWA

1. Sort a's in nondecreasing order. Let $(a_1, a_2, \ldots, a_n)$ be the resulting sequence. Let *first* := 1 and *last* := n.
2. Let δ-threshold := $\lfloor (\textit{first} + \textit{last})/2 \rfloor$. For each $i = 1, 2, \ldots, \delta$-threshold, let $e_i := d_i$ and for each $i = \delta$-threshold + 1, …, n, let $e_i := c_i$. For an n-tuple $S = (e_1, e_2, \ldots, e_n)$, evaluate $\delta_{S_{\delta\text{-threshold}}}$ and $\delta_{S_{(\delta\text{-threshold}+1)}}$.
3. If $\delta_{S_{\delta\text{-threshold}}} > 0$ and $\delta_{S_{(\delta\text{-threshold}+1)}} \leqslant 0$ then $L = f_L(e_1, e_2, \ldots, e_n)$ and goto Step 4; otherwise execute the following step.
   3.1. If $\delta_{S_{\delta\text{-threshold}}} > 0$, then *first* := δ-threshold + 1; otherwise *last* := δ-threshold, and goto Step 2.
4. Sort b's in nondecreasing order. Let $(b_1, b_2, \ldots, b_n)$ be the resulting sequence. Let *first* := 1 and *last* := n.
5. Let ζ-threshold := $\lfloor (\textit{first} + \textit{last})/2 \rfloor$. For each $i = 1, 2, \ldots, \zeta$-threshold, let $e_i := c_i$ and for each $i = \zeta$-threshold + 1, …, n, let $e_i := d_i$. For an n-tuple $S = (e_1, e_2, \ldots, e_n)$, evaluate $\zeta_{S_{\zeta\text{-threshold}}}$ and $\delta_{S_{(\zeta\text{-threshold}+1)}}$.
6. If $\zeta_{S_{\zeta\text{-threshold}}} > 0$ and $\zeta_{S_{(\zeta\text{-threshold}+1)}} \leqslant 0$ then $U = f_U(e_1, e_2, \ldots, e_n)$ and stop; otherwise execute the following step.
   6.1. If $\zeta_{S_{\zeta\text{-threshold}}} > 0$, then *first* := ζ-threshold + 1; otherwise *last* := ζ-threshold, and goto Step 5.

Theorem 5. *The EFWA algorithm finds* $\min f_L(w_1, w_2, \ldots, w_n)$ *and* $\max f_U(w_1, w_2, \ldots, w_n)$. *The EFWA algorithm requires* $O(n \log n)$ *comparisons and arithmetic operations.*

Proof. Correctness of the algorithm follows immediately from Lemma 3 and Theorem 4.

For the complexity, sorting (Steps (1) and (4)) can be done in $O(n \log n)$ comparisons [1]. Steps (2)–(3) and (5)–(6) are repeated at most $\lfloor \log n \rfloor + 1$ times [1]. Since each evaluation of δ-threshold, ζ-

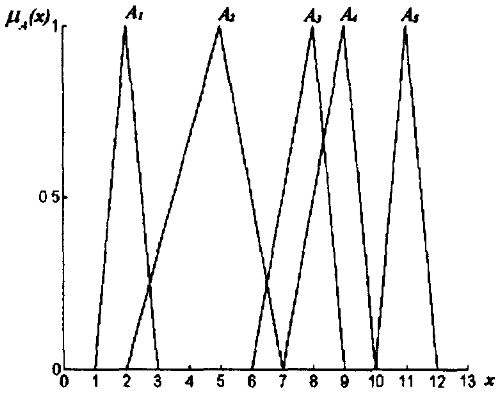

Fig. 1. Fuzzy numbers A_1, A_2, A_3, A_4 and A_5.

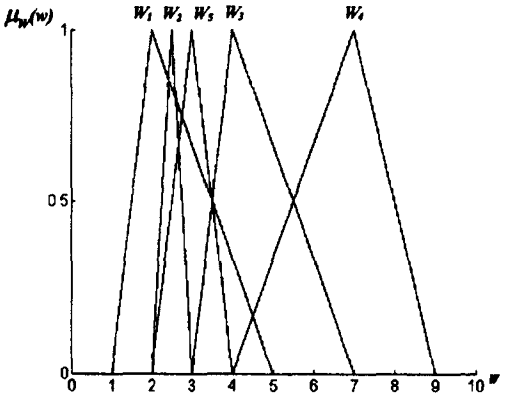

Fig. 2. Fuzzy numbers W_1, W_2, W_3, W_4 and W_5.

threshold, f_L and f_U requires $O(n)$ arithmetic operations, the algorithm requires $O(n \log n)$ arithmetic operations. □

3. Example

To illustrate the EFWA procedure, we consider a five-term weighted average operation in this section. Now, consider the weighted average

$$y = f(x_1, x_2, x_3, x_4, x_5, w_1, w_2, w_3, w_4, w_5) = \frac{x_1w_1 + x_2w_2 + x_3w_3 + x_4w_4 + x_5w_5}{w_1 + w_2 + w_3 + w_4 + w_5},$$

where the fuzzy numbers are defined as (see also Figs. 1 and 2)

$$\mu_{A_1}(x_1) = \begin{cases} x_1 - 1, & 1 \leqslant x_1 < 2, \\ 3 - x_1, & 2 \leqslant x_1 \leqslant 3, \end{cases}$$

$$\mu_{A_2}(x_2) = \begin{cases} (x_2 - 2)/3, & 2 \leqslant x_2 < 5, \\ (7 - x_2)/2, & 5 \leqslant x_2 \leqslant 7, \end{cases}$$

$$\mu_{A_3}(x_3) = \begin{cases} (x_3 - 6)/2, & 6 \leqslant x_3 < 8, \\ 9 - x_3, & 8 \leqslant x_3 \leqslant 9, \end{cases}$$

$$\mu_{A_4}(x_4) = \begin{cases} (x_4 - 7)/2, & 7 \leqslant x_4 < 9, \\ 10 - x_4, & 9 \leqslant x_4 \leqslant 10, \end{cases}$$

$$\mu_{A_5}(x_5) = \begin{cases} x_5 - 10, & 10 \leqslant x_5 < 11, \\ 12 - x_5, & 11 \leqslant x_5 \leqslant 12, \end{cases}$$

$$\mu_{W_1}(w_1) = \begin{cases} w_1 - 1, & 1 \leqslant w_1 < 2, \\ (5 - w_1)/3, & 2 \leqslant w_1 \leqslant 5, \end{cases}$$

$$\mu_{W_2}(w_2) = \begin{cases} (w_2 - 2)/0.5, & 2 \leqslant w_2 < 2.5, \\ (3 - w_2)/0.5, & 2.5 \leqslant w_2 \leqslant 3, \end{cases}$$

$$\mu_{W_3}(w_3) = \begin{cases} (w_3 - 4)/3, & 4 \leqslant w_3 < 7, \\ (9 - w_3)/2, & 7 \leqslant w_3 \leqslant 9, \end{cases}$$

$$\mu_{W_4}(w_4) = \begin{cases} w_4 - 3, & 3 \leqslant w_4 < 4, \\ (7 - w_4)/3, & 4 \leqslant w_4 \leqslant 7, \end{cases}$$

$$\mu_{W_5}(w_5) = \begin{cases} w_5 - 2, & 2 \leqslant w_5 < 3, \\ 4 - w_5, & 3 \leqslant w_5 \leqslant 4. \end{cases}$$

We choose two values for α, viz., 0 and 1. For $\alpha = 0$, the intervals of x_i and w_i are

$[a_1 = 1, b_1 = 3], [a_2 = 2, b_2 = 7],$

$[a_3 = 6, b_3 = 9], [a_4 = 7, b_4 = 10],$

$[a_5 = 10, b_5 = 12],$

$[c_1 = 1, d_1 = 5], [c_2 = 2, d_2 = 3],$

$[c_3 = 4, d_3 = 9], [c_4 = 3, d_4 = 7], [c_5 = 2, d_5 = 4],$

respectively for $i = 1, 2, \ldots, 5$.

The computational procedure is as follows:

Step 1

$(a_1, a_2, a_3, a_4, a_5) = (1, 2, 6, 7, 10),$

first := 1, *last* := 5.

Step 2

δ-threshold := $\lfloor (1 + 5)/2 \rfloor = 3,$

$S = (5, 3, 9, 3, 2),$

$\delta_{S_3} =$

$$\frac{(1-6)\cdot 5+(2-6)\cdot 3+(7-6)\cdot 3+(10-6)\cdot 2}{5+3+9+3+2}$$

$$=\frac{-26}{22}=-1.1818,$$

$\delta_{S_4} =$

$$\frac{(1-7)\cdot 5+(2-7)\cdot 3+(6-7)\cdot 9+(10-7)\cdot 2}{5+3+9+3+2}$$

$$=\frac{-48}{22}=-2.1818.$$

Step 3
Since $\delta_{S_3} < 0$ and $\delta_{S_4} < 0$, execute the following step.
Step 3.1
Let *last* := 3 and goto Step 2.
Step 2

δ-threshold $:= \lfloor (1+3)/2 \rfloor = 2$,

$S = (5, 3, 4, 3, 2)$,

$\delta_{S_2} =$

$$\frac{(1-2)\cdot 5+(6-2)\cdot 4+(7-2)\cdot 3+(10-2)\cdot 2}{5+3+4+3+2}$$

$$=\frac{42}{17}=2.4706,$$

$\delta_{S_3} =$

$$\frac{(1-6)\cdot 5+(2-6)\cdot 3+(7-6)\cdot 3+(10-6)\cdot 2}{5+3+4+3+2}$$

$$=\frac{-26}{17}=-1.5294.$$

Step 3
Since $\delta_{S_2} > 0$ and $\delta_{S_3} \leqslant 0$,

$$L = f_L(c_1, c_2, d_3, d_4, d_5) = a_2 + \delta_2 = 2 + 2.4706 = 4.4706.$$

$\min f_L$ is 4.4706 and goto Step 4.
Step 4

$(b_1, b_2, b_3, b_4, b_5) = (3, 7, 9, 10, 12)$,

first := 1, *last* := 5;

Step 5

ζ-threshold $:= \lfloor (1+5)/2 \rfloor = 3$,

$S = (1, 2, 4, 7, 4)$,

$\zeta_{S_3} =$

$$\frac{(3-9)\cdot 1+(7-9)\cdot 2+(10-9)\cdot 7+(12-9)\cdot 4}{1+2+4+7+4}$$

$$=\frac{9}{18}=0.5,$$

$\zeta_{S_4} =$

$$\frac{(3-10)\cdot 1+(7-10)\cdot 2+(9-10)\cdot 4+(12-10)\cdot 4}{1+2+4+7+4}$$

$$=\frac{-9}{18}=-0.5.$$

Step 6
Since $\delta_{\zeta_3} > 0$ and $\delta_{\zeta_4} \leqslant 0$,

$$U = f_U(d_1, d_2, d_3, c_4, c_5) = b_3 + \zeta_3 = 9 + 0.5 = 9.5.$$

$\max f_U$ is 9.5 and stop.

Accordingly, the interval for $\alpha = 0$ is [4.47, 9.50], in which each point is corresponding to the end points of the triangle representing the membership function.

The process is repeated for $\alpha = 1$ and the result obtained is [7.65, 7.65] which corresponds to the center of the triangle. Consequently, with the intervals for $\alpha = 0$ and $\alpha = 1$, the resulting membership function is determined and is plotted in Fig. 3.

4. Discussion and conclusion

In the EFWA algorithm, sorting (Steps 1 and 4) can be done in $O(n \log n)$ comparisons [1]. Since Steps 2–3 and 5–6 require constant number of comparisons and are repeated at most $\lfloor \log n \rfloor + 1$ times, it needs $O(\log n)$ comparisons. Overall, the algorithm does $O(n \log n)$ comparisons in the worst case. Since each evaluation of δ-threshold,

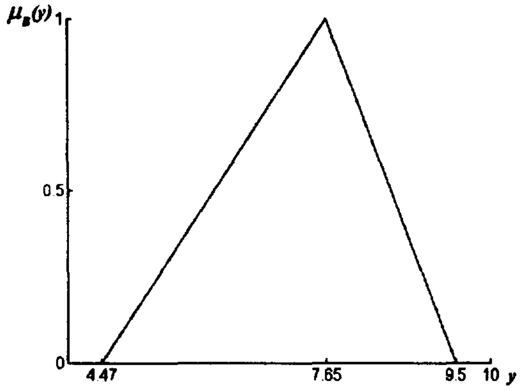

Fig. 3. Five-term fuzzy weighted average for the fuzzy numbers in Figs. 1 and 2.

ζ-threshold, f_L and f_U requires $O(n)$ arithmetic operations, the algorithm requires $O(n \log n)$ arithmetic operations. The FWA algorithm requires $O(2^n)$ comparisons and arithmetic operations [3] and the IFWA algorithm needs $O(n^2)$ comparisons and arithmetic operations [4]. This shows that the EFWA algorithm requires much less comparisons and arithmetic operations than those of the FWA algorithm and the IFWA algorithm, particularly if the number of the fuzzy number is quite large. Table 1 shows the complexity of each algorithm.

Table 1
Complexity of each algorithm

	EFWA	IFWA	FWA
Arithmetic operations	$O(n \log n)$	$O(n^2)$	$O(2^n)$
Comparisons	$O(n \log n)$	$O(n^2)$	$O(2^n)$

References

[1] S. Basse, *Computer Algorithms: Introduction to Design and Analysis* (Addison-Wesley, Reading, MA, 1991).

[2] A. Buczak and R. Uhrig, Decision fusion by fuzzy set operations, *Proc. FUZZ-IEEE* (1994) 1412–1417.

[3] W.M. Dong and F.S. Wong, Fuzzy weighted averages and implementation of the extension principle, *Fuzzy Sets and Systems* **21** (1987) 183–199.

[4] T.J. Liou and M.J.J. Wang, Fuzzy weighted average: an improved algorithm, *Fuzzy Sets and Systems* **49** (1992) 307–315.