WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia
FUZZ IEEE

A Speculative Algorithm to Extract Fuzzy Measures from Sample Data

Xiaojing Wang*, Angel F. Garcia Contreras†, Martine Ceberio‡,
Christian Del Hoyo§, and Luis C. Gutierrez¶
Computer Science Department
The University of Texas at El Paso
El Paso, Texas 79968-0518
* xwang@utep.edu, † afgarciacontreras@miners.utep.edu, ‡ mceberio@utep.edu,
§ cdelhoyo@miners.utep.edu, and ¶ lcgutierrez@miners.utep.edu

***Abstract*—In Multi-Criteria Decision Making (MCDM), decisions are based on several criteria that are usually conflicting and non-homogenously satisfied. Non-additive (fuzzy) measures along with the Choquet integral can model and aggregate the levels of satisfaction of these criteria by considering their relationships. However, in practice, it is difficult to identify such fuzzy measures. An automated process is necessary and can be used when sample data is available. Several optimization approaches have been proposed to extract fuzzy measures from sample data; for example, genetic algorithms, gradient descent algorithms, and the Bees algorithm. In this article, instead of using the search space as the primary focus of our research, we propose an algorithm that speculates on the value of the objective function before actually arriving to it. In addition, contrary to previous approaches to extracting fuzzy measures, our algorithm guarantees the solution to be global. Our experimental results show that our algorithm improves the performance of previous approaches.**

I. Introduction

Very often, decisions are based on several conflicting criteria; e.g., which car to buy that is cheap and energy efficient. This kind of complex decision process is called Multi-Criteria Decision Making (MCDM).

In general, on a daily basis, when the decision is not critical, in order to reach a decision, we mentally "average / sort" these criteria along with their satisfaction levels. In computer science, averaging corresponds to aggregating values of satisfaction with weights on each criterion, reflecting its importance in the overall score (a.k.a. additive aggregation), that is, calculating the overall score of an alternative with the weighted sum of the criterion scores. In mathematical terms, we can say that the weight assigned to different sets of criteria in the weighted average approach forms an "additive measure". Additive aggregation, however, assumes that criteria are independent, which is seldom the case [2].

Non-linear approaches also prove to lead to solutions that are not completely relevant [9].

This should change if we consider possible dependence between criteria. For example, if two criteria are strongly dependent, it means that both criteria express, in effect, the same attribute. As a result, when we consider the set consisting of these two criteria, we should assign to this set the same weight as to each of these criteria – and not double the weight as in the weighted sum approach. In general, the weights associated to different sets may not be the same as the sum of the weights associated to individual criteria. In mathematics, such non-additive functions assigning numbers to sets are known as non-additive (fuzzy) measures. It is therefore reasonable to describe the dependence between different criteria by using an appropriate non-additive (fuzzy) measure.

For such measures, in general, $m(A \cup B) \neq m(A) + m(B)$, where A and B are two disjoint sets of criteria. In particular, in the cases we consider, the total weight $m(X) = 1$ associated to the set X of all the criteria is, in general, smaller than the sum of the weights $m(\{1\}) + m(\{2\}) + \ldots$ associated to different criteria.

Now, we need to aggregate the scores x_i corresponding to different criteria by using an appropriate fuzzy measure. We can no longer add these scores with the weights $m(\{i\})$ corresponding to individual attributes, because the sum of these weights is, in general, larger than 1, so we need to decrease the weights assigned to different criteria.

One possible approach is to use the most optimistic combination; we sort the values x_i in increasing order: $x_1 \leq x_2 \leq \ldots \leq x_n$, and then try to assign as much weight as possible to larger (more optimistic) values x_i. We start with the largest value x_n that we take with the full weight $m(\{n\})$. For the next best value x_{n-1}, we select the largest weight for which the group consisting of the two best criteria is assigned its largest possible weight $m(\{n-1, n\})$. Since we already know the weight $m(\{n\})$ assigned to the n-th criterion, we can thus find the weight assigned to the next best criterion as the difference $m(\{n-1, n\}) - m(\{n\})$. Similarly, we find the weight for the $(n-2)$-nd criterion from the condition that the three best criteria gets assigned the largest possible weight $m(\{n-2, n-1, n\})$; this means that we assign, to this criterion, the weight equal to the difference $m(\{n-2, n-1, n\}) - m(\{n-1, n\})$. In general, to the i-th criterion, we assign the weight $m(\{i, i+1, \ldots, n\}) - m(\{i, \ldots, n\})$. With these weights, we get the following weighted combination:

$$\sum_{i=1}^{n} x_i \cdot (m(\{i, i+1, \ldots, n\}) - m(\{i+1, \ldots, n\})).$$

This combination is known as the Choquet integral. Choquet integrals with respect to fuzzy measures are actively used in

978-1-4673-1506-7/12/$31.00 ©2012 IEEE

Multi-Criteria Decision Making [6].

However, to make this happen, fuzzy measures need to be determined: they can either be identified by a decision maker/expert or by an automated system that extracts them from sample data. Since human expertise might not always be available and getting accurate fuzzy values (even from an expert) might be tedious [11], we focus here on extracting fuzzy measures from sample data.

The sample data that we use is a set of overall preference values (i.e., preferences that would otherwise be obtained after combining criteria satisfaction levels and an appropriate fuzzy measure through Choquet integral) associated with given inputs (i.e., items that we need to decide on, such as cars). Fuzzy measure extraction seeks to determine the fuzzy measure that, when combined in a Choquet integral, returns a value that best models the expert's decision, i.e., the corresponding sample data value from expert. This problem is therefore tackled as an optimization problem. Several optimization approaches have been used to extract fuzzy measures from sample data, such as gradient descent algorithms [4], genetic algorithms [3], [17], [19], and the Bees algorithm [18]. More specifically, fuzzy measure extraction constitutes a constrained optimization problem since the optimal solution must also satisfy the monotonicity constraints inherent to the fuzzy measure we aim at determining.

All algorithms we mentioned earlier focus on search space and use the random search to increase the possibility of getting global results. Although some of these algorithms perform global search over the search space, they cannot guarantee to get the global result. In this article, instead of focusing on search space, we propose a speculative algorithm that makes optimistic assumptions about the optimum value of the objective function before actually arriving to it. For this minimization problem, we always bet that the optimum value of the objective function lies in the lower part of the function's range and focus on this part of objective's range. The bottom line of this algorithm is that performance is not compromised if our "guess" was not correct. Besides, we use a complete interval solver to verify our speculations on the value of the objective function which is key to guarantee that the solutions we reach are global. Overall, our proposed speculative algorithm not only improves the performance of previous attempts but also guarantees the global result.

The article is organized as follows: Section II provides background and recalls necessary definitions on MCDM, fuzzy measures, fuzzy integrals, and fuzzy measure extractions. Section III introduces the optimization problem corresponding to Fuzzy Measure Extraction (FME) and recalls existing approaches to solving the specific problem of FME. We present our approach in Section IV, describe our experimental strategy, report, and analyze the results in Section V. Finally, we draw conclusions and propose directions for future work in Section VI.

II. Background

A. Multicriteria Decision Making

Multicriteria decision making (MCDM) is the making of decisions based on multiple criteria (or attributes). It is a 3-tuple problem $(X, A, \succeq)$, where:

- X is the set of consequences;
- $A = \{1, \cdots, n\}$ is the (finite) set of n criteria (or attributes); and
- $\succeq$ is a preference relation on the set of consequences.

The set of consequences X is a multidimensional space, where $X \subseteq X_1 \times \cdots \times X_n$, and each X_i represents a set of values of criterion i, where $i \in A$. For each $i \in A$, there is a preference relation $\succeq_i$ on each space X_i, such that for $x_i, y_i \in X_i$, $x_i \succeq_i y_i$ means that x_i is preferred to y_i. And there is a global preference relation $\succeq$ on X.

Note: The reason why X can be a proper subset of $X_1 \times \cdots \times X_n$ is because all combinations of all criteria values do not necessarily exist: each n-tuple of $X_1 \times \cdots \times X_n$ represents a possible instance / an alternative to pick from, all of which are not necessarily possible. For instance, consider the case of cars: one criterion being the price, another being the year of make. It is unlikely that the lowest value of the price criterion can match any high value of the year of make; *i.e., there is likely not a recent car that is very cheap.*

For example, let us continue with the MCDM case of buying a car. The decision is based on the buyer's preference on several criteria, such as, manufacturer, model, power, cost, mileage per gallon. As a result:

$$A = \{\text{manufacturer, model, power, cost, mileage per gallon}\}$$

Assume the buyer wants to choose one car from four alternatives, then X={car1, car2, car3, car4}. For each criterion $i \in A$, the buyer may order those four alternative cars based on his/her preference, e.g., for criterion A_i, car1 $\succeq$ car3 $\succeq$ car4 $\succeq$ car2; for criterion A_j, car2 $\succeq$ car4 $\succeq$ car1 $\succeq$ car3. Now, the buyer has to combine his/her preference with respect to all criteria for each alternative (each of the four cars) to obtain a global preference ranking such that the final order of the alternatives is in agreement with the buyer's partial preferences.

We assume that for each criterion $i \in A$, there exists a real valued function $u_i : X_i \to R$ such that for all x_i, $y_i \in X_i$:

$$x_i \succeq_i y_i \Longleftrightarrow u_i(x_i) \geq u_i(y_i)$$

Function u_i is called the i-th monodimensional utility function [10] and scales the values of all criteria onto a common (real) scale. Then an aggregation operator that "combines" the monodimensional utility functions needs to be used to represent the global preference, i.e., a preference over the set of consequences X: $\forall x, y \in X$, $x \succeq y$ or $y \succeq x$.

The common aggregation operator being used is a weighted sum; i.e.,

$$u(x) = \sum_{i=1}^{n} w_i u_i(x_i),$$

where w_i is the weight of each criterion, representing the importance of each criterion, and $\sum_{i=1}^{n} w_i = 1$. The best alternative is the one with the highest value of u. This approach is simple and easy to use with low complexity. However, using an additive aggregation operator assumes that all the criteria are independent. In practice, it is only seldom the case that criteria are indeed independent: often, decisions are based on several conflicting criteria and using linear additive aggregation will lead to possibly very counterintuitive decisions. Non-linear approaches also prove to lead to solutions that are not completely relevant. Therefore, using additive approach is often not good: based on our previous work [9], we choose to use non-additive approaches, i.e., fuzzy measures and integrals [2].

B. Fuzzy measures and integrals

Fuzzy measures are non-additive measures. They can be used to represent the degree of interaction of each subset of criteria [3]. In what follows, we consider a finite set of criteria $A = \{1, \cdots, n\}$.

Definition Let A be a finite set and $\mathcal{P}(A)$ the power set of A. A fuzzy measure (or a non-additive measure) defined on A is a set function $\mu : \mathcal{P}(A) \to [0, 1]$ satisfying the following axioms:

(1) $\mu(\emptyset) = 0$
(2) $\mu(A) = 1$
(3) if X, $Y \subseteq A$ and $X \subseteq Y$, then $\mu(X) \leq \mu(Y)$

Note: Fuzzy measures provide a weaker property, called monotonicity, than normal probability measures. The fuzzy measures are used to show the importance of each subset and how each subset of criteria interacts with others. Fuzzy measures are expensive to determine: for a set defined over n criteria, 2^n values of a fuzzy measure are needed because there are 2^n subsets of A. In reality, only $2^n - 2$ coefficients are needed since the values for the empty set and the full set are known (properties (1) and (2) from the above definition).

Two main integrals can be used to combine fuzzy measures.

Definition Let μ be a fuzzy measure on A. The Sugeno integral of a function $f : A \to R$ with respect to μ is defined by:

$$(S) \int f \circ \mu = \bigvee_{i=1}^{n} (f(x_{(i)}) \wedge \mu(A_{(i)}))$$

where $\vee$ is the supremum and $\wedge$ is the infimum. $\cdot_{(i)}$ indicates that the indices have been permuted so that $0 \leq f(x_{(1)}) \leq \cdots \leq f(x_{(n)}) \leq 1$, and $A_{(i)} = \{x_{(i)}, \ldots, x_{(n)}\}$.

Definition Let μ be a fuzzy measure on A. The Choquet integral of a function $f : A \to R$ with respect to μ is defined by:

$$(C) \int_A f d\mu = \sum_{i=1}^{n} (f(\sigma(i)) - f(\sigma(i-1)))\mu(A_{(i)})$$

where σ is a permutation of the indices in order to have $f(\sigma(1)) \leq \cdots \leq f(\sigma(n))$, $A_{(i)} = \{\sigma(i), \ldots, \sigma(n)\}$ and $f(\sigma(0)) = 0$, by convention.

The Sugeno and Choquet integrals are structurally similar, but essentially different in nature [5]: the Sugeno integral is based on non-linear operators and the Choquet integral is usually based on linear operators. The applications of Sugeno and Choquet integrals are also very different [10]: the Choquet integral is generally used in quantitative measurements, and a MCDM problem usually uses a Choquet integral as a representation function.

In this article, we focus on the Choquet integral.

C. Determining Fuzzy Measures

In MCDM, we would expect the decision maker to be more than likely to give the values of the fuzzy measure, but in most circumstances this is not the case. Attempts at making fuzzy measure identification easier for the decision makers have been made in [2], [14], [15].

- In [2], the authors attempt to make this task easier by only requiring the decision maker to give an interval of importance for each interaction.
- In [15], the author suggests a diamond pair-wise comparison, where the decision maker only must identify the interaction of 2 criteria using a labeled diamond. From there, the algorithm evaluates the values of the numeric weights.
- In [14], the author discusses user specified weights mixed with an interaction index denoted λ or ξ. This algorithm is applied using an online aggregation application [13].

However, in most cases, the decision maker still does not understand the interactions well enough to be able to give a good value of each fuzzy measure. This is where expert identification or fuzzy measure extraction comes into play.

In expert identification, an expert is used to giving all values of the fuzzy measures. Expert identification in most circumstances is unfeasible since in many cases, the decision maker does not have constant access to an expert. In addition, since there are $2^n - 2$ values of a fuzzy measure for a problem with n criteria expert identification, it would be too time consuming anyway to be practical [4].

As a result, instead of using an expert to provide us with the values of the fuzzy measure, we choose to extract the fuzzy measure.

III. FUZZY MEASURE EXTRACTION (FME) AND OPTIMIZATION

A. Relation Between our Problem and Optimization

For lack of an expert to provide all values of the fuzzy measure, we need seed data to give us an idea of the preferences / expert's opinions: we use sample data. We extract fuzzy measures starting from such seed data. Our objective is to determine a fuzzy measure that returns the closest values to the expert's coalition values: the values that constitute our seed data.

Let us take a look at the following situation: Our MCDM problem involves n criteria, and we have m sample data. It means that we have access to the following: m decision values $\tilde{y}_j$, $j \in \{1, \cdots, m\}$, corresponding to m alternative items

X_j. It means that if we knew the corresponding exact fuzzy measure μ, let us denote it by $\tilde{\mu}$, we would be able to compute $\tilde{y}_j$ as $(C)\int_A f d\tilde{\mu} = \sum_{i=1}^{n}(f(\sigma(i)) - f(\sigma(i-1)))\tilde{\mu}(A_{(i)})$, where f is a utility function defined on X.

Now, with our sample data, we only have access to the preference values of a subset of X. In order to have access to preference values of other alternatives in X, we need to determine μ, which is, all $2^n - 2$ values of the fuzzy measure. We are going to determine all values of μ such that the corresponding computed Choquet integral ideally equals the preference values of the sample data. In practice however, for lack of equaling the sample decision data, we aim at getting as close to them as possible. As a result, we aim at minimizing the following sum (and getting the "error" e as close to 0 as possible) [7]:

$$e = \sum_{j=0}^{m}\left(\tilde{y}_j - \sum_{i=1}^{n}(f(\sigma(i)) - f(\sigma(i-1)))\mu(A_{(i)})\right)^2 \quad (1)$$

When $e = 0$, the identified fuzzy measure μ is the exact solution of the problem: this is the ideal case. In most cases, we put up with reaching "only" an approximate optimal solution, that is, with $e \neq 0$ but close to 0. The reason for such a weaker outcome is that the sample data might not be fully consistent with one fuzzy measure, i.e., human decisions are not always consistent.

As a result, extracting a fuzzy measure is cast down to solving an optimization problem. This optimization problem is actually a constrained optimization problem since the values of μ (that the minimization process seeks) must satisfy monotonicity properties that characterize a fuzzy measure (as seen in Section II).

B. FME as a Constrained Optimization Problem

In constrained optimization, in addition to an objective function to minimize (or maximize), constraints need to be satisfied. In other words, a solution to such a problem is an element of the search space, among those that satisfy all constraints, that minimizes the objective function.

In the case of fuzzy measure extraction, fuzzy measures must be monotonic (1) and values must be between 0 and 1 (2). (1) defines the constraints of the FME problem; (2) defines the search space. For a problem with n criteria, the number of monotonicity constraints is $\sum_{k=1}^{n-2}\binom{n}{k} * (n-2)$. The fuzzy measure extraction problem is an optimization problem subject to constraints (monotonicity), which minimizes the objective function:

$$\min e = \sum_{j=0}^{m}\left(\tilde{y}_j - \sum_{i=1}^{n}(f(\sigma(i)) - f(\sigma(i-1)))\mu(A_{(i)})\right)^2,$$

where μ is the fuzzy measure to be determined and subject to constraints such as

- $0 \leq \mu_{(i)} \leq 1, \forall (i) \in \mathcal{P}(A)$
- $\mu_{(i)} \leq \mu_{(j)}$, if (i), $(j) \subseteq A$ and $(i) \subseteq (j)$

where A is a set of criteria, and $\mathcal{P}(A)$ is the power set of A. Since fuzzy measure extraction is a constrained optimization problem, the candidate solutions must be evaluated to make sure they fit the constraints.

C. Optimization Techniques used for FME

As pointed out just before, the problem we address can be modeled and solved as a constrained optimization problem. Several optimization approaches have been proposed to extract fuzzy measures, including gradient descent algorithms, genetic algorithms, neural networks, and Bees algorithm. We briefly go over them in what follows

Genetic algorithms have been successfully used to solve a number of optimization problems, including fuzzy measure extraction in [3], [17], and [19]. They show promise for extracting fuzzy measures. However, they suffer from the risk to fall into a local optimum. While mutations are part of genetic algorithms to try to avoid falling in local minima, they do not totally prevent it (especially if there are local optima that are in distant locations but have values close to the global optimum).

The gradient descent algorithm was also proposed for FME, see [4], taking advantage of the lattice structure of the coefficients of the fuzzy measure. This algorithm can reach a local optimum quickly and accurately if the initial values are properly selected. However, the monotonicity constraints need to be checked at every iteration. This algorithm was improved on in [1]. According to the experiments reported in [1], the modified gradient-descent algorithm proved to be 385 times faster than the corresponding Genetic Algorithm approach with similar or better accuracy, on the reported set of test cases.

A neural network approach for FME was proposed in [16]: the calculation of the Choquet integral was described by a neural network and the goal of this algorithm was to find a global optimum on the search space. However, the search easily falls in a local minimum.

The Bees algorithm, proposed in [12], was also used for FME in [18]. It uses bees' natural food foraging habits as a model for the exploration of the search space. The Bees algorithm combines a local and "global" search that are both based on bees natural foraging habits. Although this algorithm provided good results for FME, there was still not enough evidence to prove that the algorithm does not fall into a local optimum.

IV. Identifying Fuzzy Measures Using A Speculative Algorithm

Although previous attempts using optimization algorithms have been shown to extract fuzzy measures successfully from sample data, they all have limitations. In particular, the returned solution (found minimum of the objective function) might just be a local minimum, or even worse, a good value.

There is no guarantee that it would be the global minimum at all. Moreover, uncertainty might be part of the model to solve. It is reasonable for experts to provide data in ranges instead of precise values. Using intervals allows to take this kind of uncertainty into account. Furthermore, when dealing with problems defined on real numbers, the actual computations will round each real number to the most "relevant" floating-point number. Rounding errors can lead the returned result to be dramatically different from the original expected solution.

The work presented in this article addresses the above-mentioned issues: it guarantees results to be global, it can factor in interval data, and will not be prone to rounding errors. Our proposed approach is an interval-based technique that we called a speculative algorithm. Instead of using the search space as the primary focus of our search, the proposed algorithm speculates on the value of the objective function before actually arriving to it. The aim of this algorithm is to speed up the search process by primarily focusing on the objective function's range and always betting that the minimum value of the objective function lies in the lower part of the function's range. If our speculation is correct, then we can restrict the search space to the area that allows the objective function to take on the speculated value, and keep going to the next lower half of the range. In addition, the performance of our algorithm is not compromised: if we speculate wrong, then we do not lose more time than we would have if we had not speculated at all.

In order to verify our speculations on the range of the objective function and guarantee the global results, we use an interval solver - RealPaver. RealPaver [8] is a complete, interval-based, continuous constraint solver able to solve non-linear systems through interval computations. For the purpose of our speculative algorithm, we primarily use the narrowing function of RealPaver, which discard areas of the search space that are clearly not solution and returns a shrunk search space that contains all solutions.

Eventually, let us call e^* the optimum value of the objective function that is found by our algorithm. e^* is guaranteed to be the global minimum due to the following: our optimistic approach always bets on the lowest values and only "jump" up to larger values if RealPaver has found no feasible element in the search space matching the unrealistically speculated lower value of the objective function.

Let us now take a look at the algorithm in more details. In the case of fuzzy measure extraction, the problem is defined as follows:

- the variables are the coefficients of the fuzzy measure: for n criteria, there are $2^n - 2$ variables, and the initial domain of each variable is $[0, 1]$, which defines the initial search space as $D_0 = [0, 1]^{2^n - 2}$;
- the set of constraints is a set of inequality following the monotonicity of fuzzy measure: there are $\sum_{k=1}^{n-2} \binom{n}{k} * (n - 2)$ constraints;
- the objective function e is:

$$e = \sum_{j=0}^{m} \left(\tilde{y}_j - \sum_{i=1}^{n} (f(\sigma(i)) - f(\sigma(i-1)))\mu(A_{(i)}) \right)^2 \quad (2)$$

  The way we deal with e as a constraint sent to RealPaver is by forcing the values of e to be in a specified range R: $e \in R$.

A problem file sent to RealPaver typically consists of the above variables and constraints, in which the initial search space is updated to the current search space and the range of e is the currently speculated one. In the pseudocode that follows, we denote such a problem P by $P = D + R$ where D is the current search space and R is the current speculated range of the objective function. Our algorithm roughly unwinds as follows:

Algorithm 1 Our speculative algorithm

// Initial problem $P_0 = D_0 + R_0$
$D_1 = SendToRealPaver(P_0)$ // reduced search space
$[a_1, b_1] = e(D_1)$ //new range of the objective function e
Bisect($[a_1, b_1]$,$R_1 = [a_1, \frac{(a_1+b_1)}{2}]$,$R_2 = [\frac{(a_1+b_1)}{2}, b_1]$)
Create 2 new sub problems: $P_1 = D_1 + R_1$ and $P_2 = D_1 + R_1$
Push P_1 and P_2 into Stack S, with lower-valued P_1 at the top
while S is not empty and no solution is found **do**
    $P = D + R$ problem at the top of S
    $D_1 = SendToRealPaver(P)$ // reduced search space
    if D_1 is not empty **then**
        $[a_1, b_1] = e(D_1)$ //new range of the objective function e
        if D_1 is small enough or $[a_1, b_1]$ is small enough **then**
            $[a_1, b_1]$ is our optimum value
        else
            Bisect($[a_1, b_1]$,R_1,R_2)
            Create 2 new sub problems: $P_1 = D_1 + R_1$ and $P_2 = D_1 + R_1$
            Push P_1 and P_2 into S
        end if
    end if
end while

V. Experiments and Results

A. Testing Methodology

The goal of our experiments is to show that our speculative approach allows to improve the quality of the extracted fuzzy measure. In order to show this, we designed a fuzzy measure μ for 4 criteria, presented in Table I, and ran our algorithm to see how well it was able to reconstruct it. Besides, in order to be able to compare the performance of the speculative algorithm against existing approaches, we followed the same procedure as in [4] and [3], as described hereafter.

The input-output system contains m sample data with $Y = f_c(X) + g$, where Y is the vector of the system outputs, X is a n-tuple input $(x_1, \cdots, x_i, \cdots, x_n)$ with $x_i \in \{0, 1\}$ for n criteria, $f_c(X)$ is the calculated Choquet integral, and g is a centered gaussian noise. In the same manner as [4] and [3] did, we also checked 5 different variances, σ^2 = 0.0, 0.00096, 0.00125, 0.00625, and 0.0125 respectively. Our algorithm was executed on an Intel Xeon e5540 @2.53GHz machine. In order to find the optimal solution (most fitting fuzzy measure), we were interested in finding the minimum of the least square error E, where $E = \frac{e}{m} = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y_i})^2$. For 4 criteria, we tested with 3 different sample sizes ($m = 80, 120, 180$), and calculated the average as the final result.

TABLE I
FUZZY MEASURE TO BE IDENTIFIED FOR EXAMPLE I

A	$\mu(A)$	**A**	$\mu(A)$	**A**	$\mu(A)$
$\{1\}$	0.1	$\{1,2\}$	0.3	$\{1,2,3\}$	0.5
$\{2\}$	0.2105	$\{1,3\}$	0.3235	$\{1,2,4\}$	0.8667
$\{3\}$	0.2353	$\{1,4\}$	0.7333	$\{1,3,4\}$	0.8824
$\{4\}$	0.6667	$\{2,3\}$	0.4211	$\{2,3,4\}$	0.9474
		$\{2,4\}$	0.8070		
		$\{3,4\}$	0.8235		

B. *Quality of Solutions*

Table II shows our experimental results for 4 criteria. We compare them with the results of the Bees algorithm, genetic algorithm, and gradient descent algorithm in the same table. Results of the Bees algorithm, genetic algorithm, and gradient descent algorithm are from [18]. The final obtained fuzzy measures are in table III.

We observe that when the variance of the gaussian noise is increased, the mean square error is increased accordingly. However, the mean square error E is always less than the variance, which means our algorithm does not generate negative influence to the data. Furthermore, for all variances, our results are not only much better (closer to 0) than all of the other algorithms but also very close to the optimal results (see table I).

TABLE II
COMPARISON WITH THE OTHER ALGORITHMS

σ^2	Gradient Descent	Genetic Algorithm	Bees Algorithm	Speculative Algorithm
0.0	1.4E-7	0.00141472	1.47E-6	3.73E-9
0.00096	0.00083	0.0014723	0.000509	1.55E-5
0.00125	0.0108	0.00141241	0.001106	2.65E-5
0.00625	0.0530	0.00183267	0.004566	0.000705
0.01250	0.1054	0.00241865	0.009878	0.002898

C. *Other test examples*

We also tested for the problems with 5 criteria and 6 criteria. Table IV and table V show the fuzzy measure values to be identified for 5 criteria and 6 criteria, respectively. These values are generated randomly and follow the constraints we mentioned in Section III-B.

TABLE III
FINAL OBTAINED FUZZY MEASURE ($n = 4$)

μ	σ^2				
	0.0	0.00096	0.00125	0.00625	0.01250
$\{1\}$	0.1	0.10023	0.10071	0.10001	0.10461
$\{2\}$	0.2105	0.21011	0.20995	0.21253	0.20352
$\{3\}$	0.2353	0.2363	0.23548	0.23436	0.24098
$\{4\}$	0.6667	0.6667	0.66810	0.66488	0.66071
$\{1,2\}$	0.3	0.30006	0.29991	0.30440	0.29790
$\{1,3\}$	0.3235	0.32284	0.32389	0.32506	0.31564
$\{1,4\}$	0.7333	0.73356	0.73178	0.72729	0.74509
$\{2,3\}$	0.4211	0.42075	0.42203	0.42134	0.41666
$\{2,4\}$	0.807	0.80784	0.80511	0.80926	0.80865
$\{3,4\}$	0.8235	0.82299	0.82261	0.81920	0.82530
$\{1,2,3\}$	0.5	0.49991	0.50054	0.49681	0.50624
$\{1,2,4\}$	0.8667	0.86690	0.86715	0.86847	0.86636
$\{1,3,4\}$	0.8824	0.88203	0.88239	0.88885	0.87406
$\{2,3,4\}$	0.9474	0.94774	0.94790	0.94655	0.94404

For each problem, we added 5 different Gaussian noises ($\sigma^2 = 0.0, 0.00096, 0.00125, 0.00625, and 0.0125$), and for each noise, we generated 3 different size of samples ($m = 120, 180, 240$) and calculated the average as the final result.

Table VI shows the mean square error (E) for problems with 4, 5, and 6 criteria. Although the number of variables increases exponentially with the number of criteria (from 14 for $n = 4$ to 62 for $n = 6$), the mean square error E does not become worse, even better in some cases.

The fuzzy measures for $\sigma^2 = 0.0$ for $n = 5$ and $n = 6$ obtained by using our speculative algorithm are shown in Table VII and Table VIII, respectively. The obtained fuzzy measure exactly matches the value of actual measures.

Table IX shows the execution time for $n = 4, 5, 6$, and for different number of variables (from 14 to 62), when variance is small, the execution time for $n = 4$ is 10 times faster than the other 2 cases. However, with the increasing of the variance, the execution times for different number of variables are very close.

TABLE IV
FUZZY MEASURE TO BE IDENTIFIED FOR $n = 5$

$\mu_1 = 0.1$	$\mu_{12} = 0.3$	$\mu_{123} = 0.5$	$\mu_{1234} = 0.8$
$\mu_2 = 0.2105$	$\mu_{13} = 0.3235$	$\mu_{124} = 0.622$	$\mu_{1235} = 0.9206$
$\mu_3 = 0.2353$	$\mu_{14} = 0.4562$	$\mu_{125} = 0.8667$	$\mu_{1245} = 0.9542$
$\mu_4 = 0.4276$	$\mu_{15} = 0.733$	$\mu_{134} = 0.725$	$\mu_{1345} = 0.9614$
$\mu_5 = 0.6667$	$\mu_{23} = 0.4211$	$\mu_{135} = 0.8824$	$\mu_{2345} = 0.9822$
	$\mu_{24} = 0.48$	$\mu_{145} = 0.8546$	
	$\mu_{25} = 0.8070$	$\mu_{234} = 0.78$	
	$\mu_{34} = 0.5605$	$\mu_{235} = 0.9474$	
	$\mu_{35} = 0.8235$	$\mu_{245} = 0.825$	
	$\mu_{45} = 0.753$	$\mu_{345} = 0.8913$	

TABLE V
FUZZY MEASURE TO BE IDENTIFIED FOR $n = 6$

$\mu_1 = 0.06$	$\mu_{12} = 0.08$	$\mu_{123} = 0.3$	$\mu_{1234} = 0.73$	$\mu_{12345} = 0.99$
$\mu_2 = 0.04$	$\mu_{13} = 0.14$	$\mu_{124} = 0.32$	$\mu_{1235} = 0.5$	$\mu_{12346} = 0.89$
$\mu_3 = 0.08$	$\mu_{14} = 0.15$	$\mu_{125} = 0.34$	$\mu_{1236} = 0.69$	$\mu_{12356} = 0.85$
$\mu_4 = 0.03$	$\mu_{15} = 0.14$	$\mu_{126} = 0.46$	$\mu_{1245} = 0.57$	$\mu_{12456} = 0.97$
$\mu_5 = 0.12$	$\mu_{16} = 0.44$	$\mu_{134} = 0.32$	$\mu_{1246} = 0.88$	$\mu_{13456} = 0.84$
$\mu_6 = 0.03$	$\mu_{23} = 0.22$	$\mu_{135} = 0.26$	$\mu_{1256} = 0.8$	$\mu_{23456} = 0.91$
	$\mu_{24} = 0.26$	$\mu_{136} = 0.49$	$\mu_{1345} = 0.51$	
	$\mu_{25} = 0.14$	$\mu_{145} = 0.34$	$\mu_{1346} = 0.75$	
	$\mu_{26} = 0.33$	$\mu_{146} = 0.67$	$\mu_{1356} = 0.56$	
	$\mu_{34} = 0.14$	$\mu_{156} = 0.5$	$\mu_{1456} = 0.83$	
	$\mu_{35} = 0.19$	$\mu_{234} = 0.72$	$\mu_{2345} = 0.9$	
	$\mu_{36} = 0.22$	$\mu_{235} = 0.41$	$\mu_{2346} = 0.81$	
	$\mu_{45} = 0.26$	$\mu_{236} = 0.58$	$\mu_{2356} = 0.71$	
	$\mu_{46} = 0.17$	$\mu_{245} = 0.41$	$\mu_{2456} = 0.76$	
	$\mu_{56} = 0.41$	$\mu_{246} = 0.37$	$\mu_{3456} = 0.73$	
		$\mu_{256} = 0.48$		
		$\mu_{345} = 0.28$		
		$\mu_{346} = 0.67$		
		$\mu_{356} = 0.43$		
		$\mu_{456} = 0.61$		

TABLE VI
MEAN SQUARE ERROR FOR $n = 4$, $n = 5$ AND $n = 6$

σ^2	$n = 4$	$n = 5$	$n = 6$
0.0	3.73E-9	2.47E-5	3.47E-9
0.00096	1.55E-5	3.30E-5	6.44E-7
0.00125	2.65E-5	3.43E-5	1.13E-6
0.00625	0.0007	0.0004	5.4E-5
0.01250	0.0029	0.00143	0.0026

TABLE VII
FINAL OBTAINED FUZZY MEASURE ($n = 5$)

$\mu_1 = 0.0992$	$\mu_{12} = 0.2986$	$\mu_{123} = 0.4985$	$\mu_{1234} = 0.8001$
$\mu_2 = 0.2109$	$\mu_{13} = 0.3176$	$\mu_{124} = 0.6219$	$\mu_{1235} = 0.9359$
$\mu_3 = 0.2363$	$\mu_{14} = 0.4562$	$\mu_{125} = 0.8627$	$\mu_{1245} = 0.9558$
$\mu_4 = 0.4272$	$\mu_{15} = 0.7322$	$\mu_{134} = 0.7249$	$\mu_{1345} = 0.9621$
$\mu_5 = 0.6675$	$\mu_{23} = 0.4203$	$\mu_{135} = 0.8790$	$\mu_{2345} = 0.9835$
	$\mu_{24} = 0.4802$	$\mu_{145} = 0.8545$	
	$\mu_{25} = 0.8049$	$\mu_{234} = 0.78$	
	$\mu_{34} = 0.5604$	$\mu_{235} = 0.9359$	
	$\mu_{35} = 0.8235$	$\mu_{245} = 0.825$	
	$\mu_{45} = 0.7528$	$\mu_{345} = 0.8911$	

TABLE IX
AVERAGE EXECUTION TIME (S)

σ^2	$n = 4$	$n = 5$	$n = 6$
0.0	14.6	138.7	114.7
0.00096	27.4	137.0	140.3
0.00125	32.2	166.1	281.5
0.00625	240.0	383.1	473.3
0.01250	606.9	962.7	334.1

VI. CONCLUSION AND FUTURE WORK

In this article, we proposed an interval-based algorithm that speculates the value of the objective function and always bets that the optimum value of the objective function lies in the bottom part of the function's range. We then used RealPaver, a complete interval solver, to verify our speculations on the value of the objective function. By using RealPaver, the search space is narrowed down and the global results can be guaranted, as RealPaver always return a search space that contains all solutions that satisfy the constraints if there is solutions to the problem. By speculating on the range of the objective function, the search performance can be sped up, as those speculations whose solutions are not found by RealPaver are discarded, focusing on the speculated ranges for the objective function. In comparison with other algorithms that have been used for fuzzy measure extraction, the results of our speculative algorithm are closer to 0, showing the better quality of the results.

We know that the bottleneck of using a fuzzy measure is its exponential cost (in our case, a number of values to identify that is exponentially proportional to the number of criteria). In practice, it is reasonable to consider the relationship of any two criteria and ignore the interaction among 3 or more criteria: this constitutes a reasonable compromise between assuming independence of the criteria and handling all possible dependencies. The corresponding fuzzy measure is called a 2-additive fuzzy measure. In the future, we will apply our speculative algorithm to 2-additive fuzzy measures and apply it to real-world situations, such as software quality assessment, as we did with the Bees algorithm in the past. Furthermore, we plan to use our FME approach to help model non-expert decision-making process as well, helping

TABLE VIII
FINAL OBTAINED FUZZY MEASURE ($n = 6$)

$\mu_1 = 0.0595$	$\mu_{12} = 0.0805$	$\mu_{123} = 0.2998$	$\mu_{1234} = 0.73$	$\mu_{12345} = 0.99$
$\mu_2 = 0.04$	$\mu_{13} = 0.1401$	$\mu_{124} = 0.32$	$\mu_{1235} = 0.5$	$\mu_{12346} = 0.89$
$\mu_3 = 0.08$	$\mu_{14} = 0.15$	$\mu_{125} = 0.3399$	$\mu_{1236} = 0.69$	$\mu_{12356} = 0.8498$
$\mu_4 = 0.03$	$\mu_{15} = 0.1456$	$\mu_{126} = 0.4602$	$\mu_{1245} = 0.57$	$\mu_{12456} = 0.97$
$\mu_5 = 0.1195$	$\mu_{16} = 0.44$	$\mu_{134} = 0.3201$	$\mu_{1246} = 0.88$	$\mu_{13456} = 0.84$
$\mu_6 = 0.0299$	$\mu_{23} = 0.2201$	$\mu_{135} = 0.2597$	$\mu_{1256} = 0.7948$	$\mu_{23456} = 0.9102$
	$\mu_{24} = 0.26$	$\mu_{136} = 0.4902$	$\mu_{1345} = 0.51$	
	$\mu_{25} = 0.14$	$\mu_{145} = 0.3402$	$\mu_{1346} = 0.75$	
	$\mu_{26} = 0.33$	$\mu_{146} = 0.6703$	$\mu_{1356} = 0.5594$	
	$\mu_{34} = 0.1399$	$\mu_{156} = 0.5012$	$\mu_{1456} = 0.8271$	
	$\mu_{35} = 0.19$	$\mu_{234} = 0.72$	$\mu_{2345} = 0.8995$	
	$\mu_{36} = 0.22$	$\mu_{235} = 0.4098$	$\mu_{2346} = 0.81$	
	$\mu_{45} = 0.2585$	$\mu_{236} = 0.5801$	$\mu_{2356} = 0.71$	
	$\mu_{46} = 0.17$	$\mu_{245} = 0.4107$	$\mu_{2456} = 0.7602$	
	$\mu_{56} = 0.4104$	$\mu_{246} = 0.37$	$\mu_{3456} = 0.7304$	
		$\mu_{256} = 0.4813$		
		$\mu_{345} = 0.2798$		
		$\mu_{346} = 0.67$		
		$\mu_{356} = 0.43$		
		$\mu_{456} = 0.6096$		

understand which information factors contribute to changing decisions' outcomes.

ACKNOWLEDGMENT

The work presented here was partially supported by NSF grant CCF No. 0953339.

REFERENCES

[1] S. H. Alavi, J. Jassbi, P. J. A. Serra, and R. A. Ribeiro. Defining fuzzy measures: A comparative study with genetic and gradient descent algorithms. In *Intelligent Engineering Systems and Computational Cybernetics*, pages 427–437. Springer Netherlands, 2009.

[2] M. Ceberio and F. Modave. An interval-valued, 2-additive choquet integral for multi-criteria decision making. In *Proceedings of the 10th Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'04)*, Perugia, Italy, July 2004.

[3] E. F. Combarro and P. Miranda. Identification of fuzzy measures from sample data with genetic algorithms. *Computers & Operations Research*, 33(10):3046–3066, 2006.

[4] M. Grabisch. A new algorithm for identifying fuzzy measures and its application to pattern recognition. In *Proceedings of 4th IEEE International Conference on Fuzzy Systems*, Yokohama , Japan, March 1995.

[5] M. Grabisch. The application of fuzzy integrals in multicriteria decision making. *European Journal Of Operational Research*, 89(3):445–456, 1996.

[6] M. Grabisch, I. Kojadinovic, and P. Meyer. A review of methods for capacity identification in choquet integral based multi-attribute utility theory applications of the kappalab r package. *European Journal of Operational Research*, 186(2):766–785, 2008.

[7] M. Grabisch, H.T. Nguyen, and E. A. Walker. *Fundamentals of uncertainty calculi with applications to fuzzy inference*. Kluwer Academic Publishers, Norwell, MA, 1994.

[8] L. Granvilliers and F. Benhamou. Realpaver: An interval solver using constraint satisfaction techniques. *ACM Transactions on Mathematical Software (TOMS)*, 32(1):138–156, 2006.

[9] F. Modave, M. Ceberio, and V. Kreinovich. Choquet integrals and owa criteria as a natural (and optimal) next step after linear aggregation: A new general justification. In *Proceedings of MICAI'2008*, pages 741–753, 2008.

[10] F. Modave and P. W. Eklund. A measurement theory perspective for mcdm. In *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, pages 1068–1071, Melbourne, Australia, 2001.

[11] T. Mogoc and V. Kreinovich. How to relate fuzzy and owa estimates. In *North American Fuzzy Information Processing Society (NAFIPS'2010)*.

[12] D. Pham, A. Ghanbarzadeha, E. Koc, S.Otri, S. Rahim, and M. Zaidi. The bees algorithm-a novel tool for complex optimization problems. In *Proceedings of 2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2006)*, pages 454–459, 2006.

[13] E. Takahagi. Usage: Fuzzy measure-Choquet integral calculation system (λ fuzzy measure and sensitivity analysis). http://www.isc.senshu-u.ac.jp/~thc0456/Efuzzyweb/mant2/mant2.html.

[14] E. Takahagi. On identification methods of λ -fuzzy measures using weights and λ. *Japanese Journal of Fuzzy Sets and Systems*, 12(5):665–676, 2000.

[15] E. Takahagi. A fuzzy measure identification method by diamond pairwise comparisons and ϕ_s transformation. *Fuzzy Optimization and Decision Making*, 7(3):219–232, 2008.

[16] J. Wang and Z. Wang. Using neural networks to determine sugeno measures by statistics. *Neural Networks*, 10(1):183–195, 1997.

[17] W. Wang, Z. Wang, and G. J. Klir. Genetic algorithms for determining fuzzy measures from data. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 6(2):171–183, 1998.

[18] X. Wang, J. Cummins, and M. Ceberio. The bees algorithm to extract fuzzy measures for sample data. In *North American Fuzzy Information Processing Society (NAFIPS'2011)*, El Paso, TX, March 2011.

[19] Z. Wang, K. Leung, and J. Wang. A genetic algorithm for determing nonadditive set functions in information fusion. *Fuzzy Sets and Systems - Special issue on fuzzy measures and integrals*, 102(3):463–469, 1999.