An Intuitionistic Fuzzy Methodology for Component-Based Software Reliability Optimization

HENRIK MADSEN¹, GRIGORE ALBEANU²*, and FLORIN POPENŢIU-VLĂDIEESCU³

¹DTU Informatics, Richard Petersens Plads, 2800, Lyngby, Denmark
²DTU Informatics, Richard Petersens Plads, 2800, Lyngby, Denmark
³City University London, UNESCO Chair in Information and Communication Engineering, London, U.K.; University of Oradea, UNESCO Department (Head of), Oradea, Romania

(Received on Dec. 18, 2010, revised on August 18, 2011)

Abstract: Component-based software development is the current methodology facilitating agility in project management, software reuse in design and implementation, promoting quality and productivity, and increasing the reliability and performability. This paper illustrates the usage of intuitionistic fuzzy degree approach in modelling the quality of entities in imprecise software reliability computing in order to optimize management results. Intuitionistic fuzzy optimization algorithms are proposed to be used for complex software systems reliability optimization under various constraints.

Keywords: Software reliability optimization, multi-objective optimization, intuitionistic fuzzy numbers, Monte Carlo method

1. Introduction

In this paper the software reliability optimization problem is considered when complex software architectures are used. The current methodology is component-based and facilitates the agility in software management, component reuse both in design and implementation, promotes quality, productivity, and contributes to performability increasing. For the aim of this paper only the reliability characteristic are considered.

As mentioned in [1], “software assets, or components, include all software products, from requirements and proposals, to specifications and design, to user manuals and test suites.” Moreover, in [2] it is considered that the software analysis depends on three assets: code, specification, and test received. However, to establish clearly the target of our paper, by software component we refer to any of the following assets: functions (belonging to a reusable library), modules (or classes), libraries (collections of reusable functions), packages (collections of reusable classes), and applications (programs to be reused by “exec” service, including code, files, and databases) when these assets follow the independence requirement.

From viewpoint of software methodologies paradigms, an important step toward efficient reuse of software components consists of the evolution from reusable function libraries to object-oriented class libraries reaching a new stage called: reusable application framework and platform. The reliability of every complex system depends on the reliability of every component and the architectural model.
Following the standard model, if the system consists of n components with reliabilities $R_j$, $j = 1, 2, \ldots, n$, and an execution path is given, for instance: 1, 3, 2, 3, 2, 1, 4, 2, n, then the path reliability is $R_1 \times R_3 \times R_2 \times R_3 \times R_2 \times R_1 \times R_4 \times R_2 \times R_n$. The system reliability can be estimated by averaging over all path reliabilities, as shown in [3, 4]. A probabilistic approach based on the probabilities of using the components will be described in the next section. The approach is a special case of component based statistical software reliability analysis.

The most natural approach considers that the reliability characteristics are independent of the software age. However, the performance of the software would decrease in time due to inappropriate management of the data structures (files, lists, trees and graphs, dynamic arrays). More aspects related to software aging are given in [5]. Therefore, the software without periodically maintenance, or rejuvenation, would loose fractions of initial running speed. This will affect the quality of the software from the customer point of view.

An important objective when one models the software reliability is linked to behaviour prediction. The difficulty appears when consider the deployment context. As shown in [6], "the design and implementation faults of software have a different impact on the reliability of the software, depending on how frequently the faulty code is executed". If such probabilities can be estimated, the methodology mentioned above can be used. Another aspect in component-based software engineering is related to component dependencies. The exact properties of some components are not known until deployment. The class of unknown components includes: *operating system(s) modules, middleware, and network and transport services*. The usage profile can be modelled with Markov chains, but this approach is not considered here.

During software project management and, therefore, during software life cycle, many decisions are taken using imprecise data related to cost estimation (including human resources and expertise) and target reliability. Not only the lack of information, but also the hesitation, is responsible for fuzzy formulation of different aspects. This motivates the usage of fuzzy set, and intuitionistic fuzzy concept in software reliability management [7].

This paper illustrates the usage of intuitionistic fuzzy degree approach in modelling the quality of entities in imprecise software reliability computing in order to optimize management results. Firstly, the probabilistic reliability model is described related to component-based software development methodology. Then, new formulas and interpretations are given in the context of intuitionistic fuzzy paradigm introduced by Atanassov [8]. Combined Monte-Carlo [9], Mahapatra [10, 11], and standard optimization methods [12] are proposed to be applied for software reliability optimization problems considered under various constraints. Finally, a discussion related to the practical usage of the proposed methods and conclusions are provided.

**Notation**

- $n$: number of software components
- $m$: number of operations
- $\gamma$: the target of the system reliability ($0 < \gamma < 1$)
- $T$: the length of the mission interval ($T > 0$)
- $\lambda_j$: the failure intensity of the $j$th component
- $p_i$: the probability of executing the operation $i$ ($1 \leq i \leq m$)
- $\tau_j$: the expected proportion of the total mission that the software spends executing in component $j$ ($1 \leq j \leq n$)
An Intuitionistic Fuzzy Methodology for Component-Based Software Reliability Optimization

2. The Probabilistic Reliability Modelling

The basic structure of any sequential algorithm is the sequence. The statements address computing aspects (call statements, expression evaluations, etc.), branching (if, case/switch) and looping mechanisms (for, while, do {} while, repeat {} until). Any sequential computer program will have an execution path described by a regular expression, expanded in a linear sequence during run-time.

Let \( n \) be the number of software components, \( \gamma \) the target of the system reliability (0 < \( \gamma \) < 1) and \( T \) be the length of the mission interval (\( T > 0 \)). The time interval \([0, T]\) will be considered during analysis.

When no rejuvenation is considered, the exponential model is a good assumption. For all \( j = 1, 2, \ldots, n \), let \( \lambda_j \) be the failure intensity of the \( j^{th} \) component. The probability density function of the random variable giving the time to failure of the \( j^{th} \) component is

\[
f_j(t) = \lambda_j \exp(-\lambda_j t),
\]

with the corresponding reliability function \( R_j(t) = \exp(-\lambda_j t) \). Let us denote by \( R(\lambda; t) \) the reliability of the \( j^{th} \) component: \( R(\lambda; t) = \exp(-\lambda t) \).

For a serial composition, the reliability of the system can be computed as:

\[
R(\lambda_1, \lambda_2, \ldots, \lambda_n; t) = \prod_{j=1}^{n} R(\lambda_j; t).
\]

If information concerning the operational profile is considered, \( T \) is the time mission, \( p_i \) is the probability of executing the operation \( i \), and \( \phi_{ij} \) is the time allocated to the component \( j \), then the expected proportion of the total mission time that the software spends executing in component \( j \), denoted by \( \tau_j \), is given by

\[
\tau_j = \sum_{i=1}^{n} p_i \phi_{ij}.
\]

Therefore, the reliability of the \( j^{th} \) component with respect to the proportion of time it runs is:
\[ R(\lambda; T) = \exp(-\sum_{j=1}^{n} p_j \phi_j \lambda, T) = \exp(-\tau_j \lambda, T) \]

and the reliability of the integrated system with respect to time mission \( T \) becomes \([7]\):

\[ R(\lambda_1, \lambda_2, \ldots, \lambda_n; T) = \exp(-\sum_{j=1}^{n} \sum_{i=1}^{n} p_{ij} \phi_j \lambda, T) . \]

Let \( T_C(\lambda_1, \lambda_2, \ldots, \lambda_n) \) be the total cost of software development and testing in order to achieve the failure intensities \( \lambda_1, \lambda_2, \ldots, \lambda_n \). The most used models in cost estimation are the COCOMO model \([13]\) and the Putnam’s model \([14]\). The basic COCOMO model computes software development effort (and cost) as a function of program size (expressed in estimated thousands of lines of code). Intermediate and detailed COCOMO models were proposed in order to capture supplementary aspects during software development: 1) Component related: complexity of the product, required software reliability, size of application database, the user-software interface level; Platform related: run-time performance constraints, memory constraints, etc.; Software team related: analyst capability, software engineering capability, programmer experience, programming language relevance, etc.; Project management related: use of software tools, application of software engineering methods, required development schedule.

The simplest approach can use a separable form of the total cost function:

\[ T_C(\lambda_1, \lambda_2, \ldots, \lambda_n) = \sum_{i=1}^{n} C(\lambda_i) , \]

where \( C(\lambda_i) \) is the cost of developing the component \( i \).

Let \( \Lambda = \{ (\lambda_1, \lambda_2, \ldots, \lambda_n) \in \mathbb{R}^n \mid \lambda_i \geq 0, i = 1, 2, \ldots, n \} \) be the space of parameters. The minimization of the total cost of achieving the target reliability \( \gamma \) it is written as:

\[ \begin{align*}
\min & T_C(\lambda) \\
\text{s.t.} & R(\lambda; T) \geq \gamma \\
& \lambda \in \Lambda
\end{align*} \]

with the notations \( g_0(\lambda) = \gamma \ln(\gamma) - \ln R(\lambda, T) \), and \( g_j(\lambda) = -\lambda_j, j = 1, 2, \ldots, n \), the above model can be written in the standard form for the nonlinear optimization theory \([12]\). An alternative allocation model considers the maximization of the system reliability subject to budget constraints:

\[ \begin{align*}
\max & R(\lambda; T) \\
\text{s.t.} & T_C(\lambda) \leq B \\
& \lambda \in \Lambda
\end{align*} \]

where \( B \) is the maximum cost allowed.

In order to solve the above models any optimization method (deterministic or stochastic) can be used depending on the properties of the functions appearing into the models. In the following section we consider the intuitionistic fuzzy environment and use the triangular intuitionistic fuzzy numbers for reliability computation. Finally, a methodology dealing with nonlinear optimization under imprecise objective and constraint functions is described.
3. **Intuitionistic Fuzzy Software Reliability Modelling**

Zadeh [15] introduced the fuzzy sets as a new paradigm for imprecise modelling. Then, the intuitionistic fuzzy sets (IFS) were considered by Atanassov [8], and Chen [16] represented a generalized fuzzy numbers (GFN) by a 5-tuple \((a, b, c, d; w)\) of five real numbers, such that \(0 \leq a < b < c < d\), by a membership function having \(w\) as maximum value. Mainly, the GFN denoted by \(A\) is a fuzzy subset of the real line \(\mathbb{R}\), whose degree of membership (resp. non-membership) of \(x\) in \(A\), with \(0 \leq \mu_A(x) \leq 1\), is characterized by a membership function \(\mu_A\) and a non-membership function \(\nu_A\) on \(\mathbb{R}\). Then, the intuitionistic fuzzy sets (IFS) were considered by Atanassov [8], and Chen [16].

For the universe of discourse denoted by \(X\), an intuitionistic fuzzy set (IFS) \(A\) in \(X\) is characterized by a membership function \(\mu_A(x)\) and a non-membership function \(\nu_A(x)\) on \(X\), where \(\mu_A \colon X \to [0, 1]\), and \(\nu_A \colon X \to [0, 1]\). For each point \(x\) in \(X\), \(\mu_A(x)\) (resp. \(\nu_A(x)\)) is the degree of membership (resp. non-membership) of \(x\) in \(A\), with \(0 \leq \mu_A(x) + \nu_A(x) \leq 1\). The intuitionistic fuzzy set becomes a fuzzy set if \(\nu_A(x) = 0\) for all \(x\) in \(A\).

The operations on IFS can be introduced according to the generalized fuzzy set theory, as shown, shortly, below: 1) \(A \cap B = \{(x, \min(\mu_A(x), \mu_B(x)), \max(\nu_A(x), \nu_B(x))), x \in X\}\); 2) \(A \cup B = \{(x, \max(\mu_A(x), \mu_B(x)), \min(\nu_A(x), \nu_B(x))), x \in X\}\); 3) \(A \setminus B = \{(x, \mu_A(x) - \mu_B(x)), \nu_A(x) - \nu_B(x)), x \in X\}\); 4) \(AB = \{(x, \mu_A(x)\mu_B(x)), \nu_A(x)\nu_B(x)), x \in X\}\).

The most used IFSs are the **Triangular Intuitionistic Fuzzy Numbers** (TIFNs). The TIFN \(A\) is described by five real numbers \((a_1, a_2, a_3; a', a'')\), \(a' \leq a_1 \leq a_2 \leq a_3 \leq a''\), and two triangular functions:

\[
\mu_A(x) = \begin{cases} 
\frac{x - a_1}{a_2 - a_1}, & \text{for } a_1 \leq x \leq a_2 \\
\frac{a_2 - x}{a_3 - a_2}, & \text{for } a_2 \leq x \leq a_3 \\
0, & \text{otherwise},
\end{cases}
\]

and

\[
\nu_A(x) = \begin{cases} 
\frac{a_3 - x}{a_2 - a_3}, & \text{for } a_3 \leq x \leq a_2 \\
\frac{a' - a_1}{a'' - a_2}, & \text{for } a_2 \leq x \leq a'' \\
1, & \text{otherwise}.
\end{cases}
\]

An arithmetic operation, denoted generically by \(*\), on two IFSs is a mapping of an input subset of \(\mathbb{R} \times \mathbb{R}\) (with elements \(x = (x_1, x_2)\)) onto an output sub-set of \(\mathbb{R}\) (with elements denoted by \(y\)). Let \(A_1\) and \(A_2\) be two IFSs, and \((A_1 \ast A_2)\) the resultant of the operation \(*\). Then:

\[
(A_1 \ast A_2)(y) = \left\{ \begin{array}{ll}
y, & \forall x_1, x_2, [A_1(x_1) \ast A_2(x_2)], x \in R^2, \\
\land_{y=x_1\cdot x_2} [A_1(x_1) \ast A_2(x_2)], & \forall x_1, x_2, y \in R^2, \\
\land_{y=x_1\cdot x_2} [A_1(x_1) \ast A_2(x_2)], & \forall x_1, x_2, y \in R^2,
\end{array} \right.
\]

with

\[
\mu_{A_1 \ast A_2}(y) = \land_{y=x_1\cdot x_2} [A_1(x_1) \ast A_2(x_2)], \quad \nu_{A_1 \ast A_2}(y) = \land_{y=x_1\cdot x_2} [A_1(x_1) \ast A_2(x_2)].
\]

In general, the arithmetic operations on IFSNs can be defined using the \((\alpha, \beta)\) - cuts method. Let \(\alpha, \beta \in [0, 1]\) be fixed numbers such that \(\alpha + \beta \leq 1\). A set of \((\alpha, \beta)\) - cut generated by an IFS \(A\) is defined by \(A_{\alpha, \beta} = \{(x, \mu_A(x), \nu_A(x)), x \in X, \mu_A(x) \geq \alpha, \nu_A(x) \leq \beta\}\). The \((\alpha, \beta)\) – cut of a TIFN is given by \(A_{\alpha, \beta} = \{(A_\alpha, A_\beta), [A_\alpha, A_\beta]\}\).
where:

- $A_1(\alpha)$ and $A_2(\beta)$ are continuous, monotonic increasing functions of $\alpha$, respective $\beta$;
- $A_3(\alpha)$ and $A_4(\beta)$ are continuous, monotonic decreasing functions of $\alpha$, respective $\beta$;
- $A_1(1)=A_2(1)$, and $A_1'(0)=A_2'(0)$.

When the 5-tuple notation is used, the following results are obtained:

$$\begin{align*}
\alpha(A_1) &= a_1 + \alpha(a_2 - a_1), \\
\alpha(A_2) &= a_3 - \alpha(a_3 - a_2), \\
\beta(A_1) &= a_z + \beta(a' - a_z), \\
\beta(A_2) &= a_z + \beta(a'' - a_z).
\end{align*}$$

To fulfill the aim of this paper we need the following properties, obtainable by $(\alpha, \beta)$-cuts method [10,11]: (1) If TIFN $A = (a_1, a_2, a_3; a', a'')$, and $k > 0$, then the TIFN $kA$ is given by $(ka_1, ka_2, ka_3; ka', ka''$); (2) If TIFN $A = (a_1, a_2, a_3; a', a'')$, and $k < 0$, then the TIFN $kA$ is given by $(ka_3, ka_2, ka_1; ka'', ka')$; (3) If $A = (a_1, a_2, a_3; a', a'')$ and $B = (b_1, b_2, b_3; b', b'')$ are TIFNs, then the sequence defined by $(a_1+b_1, a_2+b_2, a_3+b_3; a'+b', a''+b'')$ describes the TIFN denoted by $A \oplus B$, and the TIFN denoted by $A \otimes B$ is described by $(a_1b_1, a_2b_2, a_3b_3; a'b', a''b')$.

Let $S$ be a software system integrating a number of components according to a serial architecture. This is the case of single processor computer systems running the software during a period of time $T$. For simplicity reason we assume that every component is called only once, but this is not a serious constraint. If $R_j$ is the intuitionistic fuzzy reliability of the $j$th component, $R_S$ is the intuitionistic fuzzy reliability of the entire system (with $n$ items), and $R_j = (r_{j1}, r_{j2}, r_{j3}; r_j', r_j'')$, then $R_S = R_1 \otimes R_2 \otimes \cdots \otimes R_n$ is described by $(r_1, r_2, r_3; r', r'')$, with:

$$r_j = \prod_{i=1}^{n} r_{ji}, \quad i = 1, 2, 3, \quad r' = \prod_{j=1}^{n} r_j, \quad \text{and} \quad r'' = \prod_{j=1}^{n} r_{jj}.'$$

If $S$ is a software system composed of $n$ items running in parallel, using the above notations, we evaluate the intuitionistic fuzzy reliability of $S$ by $R_S = \Theta \prod_{j=1}^{n} (1 \Theta R_j)$, which is defined by $(r_1, r_2, r_3; r', r'')$, with

$$r_j = 1 - \prod_{i=1}^{n} (1 - r_{ji}), \quad i = 1, 2, 3; \quad r' = 1 - \prod_{j=1}^{n} (1 - r_j), \quad \text{and} \quad r'' = 1 - \prod_{j=1}^{n} (1 - r_j').$$

The above methodology permits to derive intuitionistic fuzzy reliability formulas hybrid software architectures.

4. An Intuitionistic Fuzzy Optimization Methodology

Considering the intuitionistic fuzzy environment described above, and the optimization problem defined by objectives and constraints which are formulated in a generalized fuzzy approach, let us consider the equation (1) in intuitionistic fuzzy interpretation. Firstly, the nonlinear optimization problem is considered in the most general form ($m = n+1$, the $m$th constraint is related to the cost-budget inequality):
The intuitionistic fuzzy optimization methodology is based on the following steps:

**The first step**: Using a Monte-Carlo approach, an approximate lower bound (minimum) \(L_a\) and an approximate upper bound (maximum) \(U_a\) can be obtained. According to the intuitionistic fuzzy principle, the degree of non-membership (rejection) and degree of membership (acceptance) are considered so that the sum of both values is less than one. Let \(L_r\) (resp. \(U_r\)) be the lower (resp. upper) bound of the objective function such that \(L_a \leq L_r \leq U_r \leq U_a\). It is known (theorem 7.2.1 [10,11]) that "for objective function of maximization problem, the upper bound for non-membership function is always less than that of the upper bound of membership functions." It is possible, for a given \(\theta\) (the decision maker choice) to take \(U_r = U_a - \theta\), and \(L_r = L_a\).

**The second step**: Define the membership and non-membership functions for the imprecise objective function. Various models can be used. Below is presented a simple one:

\[
\mu_{g_j}(\lambda) = \begin{cases} 
1, & \text{for } g_j(\lambda) \leq \xi_j \\
\frac{\xi_j - g_j(\lambda)}{\xi_j}, & \text{for } 0 \leq g_j(\lambda) \leq \xi_j \\
0, & \text{for } g_j(\lambda) \geq \xi_j 
\end{cases}
\]

\[
\nu_{g_j}(\lambda) = \begin{cases} 
0, & \text{for } g_j(\lambda) \leq \xi_j - \psi_j \\
\frac{\psi_j}{\psi_j - \xi_j}, & \text{for } \xi_j - \psi_j \leq g_j(\lambda) \leq \psi_j \\
1, & \text{otherwise.}
\end{cases}
\]

**The third step**: Define the membership and non-membership functions for the constraint equations. Let \(j\) be the constraint index (the model (2)), \(\xi_j\) - the tolerance for the degree of constraint acceptance, and \(\psi_j\) – the tolerance for the degree of constraint rejection (\(\psi_j = \rho \xi_j\), where \(0 < \rho < 1\)). A proposal for the membership and non-membership functions for the constraint with index \(j\) follows (\(j = 1, 2, \ldots, m\)):

\[
\mu_{g_j}(\lambda) = \begin{cases} 
1, & \text{for } g_j(\lambda) \leq \xi_j \\
\frac{\xi_j - g_j(\lambda)}{\xi_j}, & \text{for } 0 \leq g_j(\lambda) \leq \xi_j \\
0, & \text{for } g_j(\lambda) \geq \xi_j 
\end{cases}
\]

\[
\nu_{g_j}(\lambda) = \begin{cases} 
0, & \text{for } g_j(\lambda) \leq \xi_j - \psi_j \\
\frac{\psi_j}{\psi_j - \xi_j}, & \text{for } \xi_j - \psi_j \leq g_j(\lambda) \leq \psi_j \\
1, & \text{otherwise.}
\end{cases}
\]

**The fourth step**: The intuitionistic fuzzy version of the model (2) becomes a multi-objective optimization problem:

Maximize \{\(\mu(f(\lambda)), \mu_1(g_1(\lambda)), \mu_2(g_2(\lambda)), \ldots, \mu_m(g_m(\lambda))\}\}

Minimize \{\(\nu(f(\lambda)), \nu_1(g_1(\lambda)), \nu_2(g_2(\lambda)), \ldots, \nu_m(g_m(\lambda))\}\}

Subject to

\[
\mu(f(\lambda)) + \nu(f(\lambda)) \leq 1; \\
\mu_j(g_j(\lambda)) + \nu_j(g_j(\lambda)) \leq 1, j=1, 2, \ldots, m; \\
\mu_j(g_j(\lambda)) \geq \nu_j(g_j(\lambda)); \\
\mu_j(g_j(\lambda)) \geq \nu_j(g_j(\lambda)), j=1, 2, \ldots, m; \\
\nu(f(\lambda)) \geq 0; \\
\nu_j(g_j(\lambda)) \geq 0, j=1, 2, \ldots, m; \\
\lambda \in \Lambda.
\]

**The fifth step**: Transform the above problem to obtain a bi-objective optimization programming problem, like in the following model (based on equal importance assumption):
Maximize $\mu_S(\lambda) = \mu(f(\lambda)) + \mu_1(g_1(\lambda)) + \ldots + \mu_m(g_m(\lambda))$

Minimize $\nu_S(\lambda) = \nu(f(\lambda)) + \nu_1(g_1(\lambda)) + \ldots + \nu_m(g_m(\lambda))$

Subject to

$\mu(f(\lambda)) + \nu(f(\lambda)) \leq 1$

$\mu_j(g_j(\lambda)) + \nu_j(g_j(\lambda)) \leq 1, j=1,2,\ldots,m$

$\mu(f(\lambda)) \geq \nu(f(\lambda))$

$\nu(f(\lambda)) \geq 0$

$\nu_j(g_j(\lambda)) \geq 0, j=1,2,\ldots,m$

$\lambda \in \Lambda$

If unequal importance case is considered then some weights (degrees of importance) and weighted sums will be used when modelling $\mu_S(\lambda)$ and $\nu_S(\lambda)$.

The sixth step: Finally, a global optimization method (including Monte Carlo) can be used to find the solution.

The above methodology can be used for general optimization problems formulated under imprecise assumptions. Optimum testing-effort allocation problems for software developed under component-based paradigm, proposed in [3] or by Lo et al. [4] can be solved in an intuitionistic fuzzy environment, when fuzzy testing assumptions is considered. However, if the problem is formulated under precise or controlled assumptions, classical optimization methods will be used.

The proposed methodology was applied for two projects developed by recent software development methodologies. The first project is DISTeFAX [17]: a distributed faxing system designed and implemented as a secure software system facilitating the processing of meta-faxes (multipart documents obtained from individual files generated by different applications like text editors, spread sheet processors, image processing applications, etc.). The second project belongs to the e-learning field [18]. The software quality management team appreciates the information based on both membership and non-membership functions and the results obtained by the proposed optimization method.

5. Conclusions

In this paper we considered subjects like software reliability estimation based on user profile, and optimal decision related to cost-reliability models in classical and intuitionistic fuzzy environments. Based on probabilistic thinking and statistical testing we describe a reliability allocation strategy under cost effort constraints. Two problems were considered: maximize reliability under budget limitations, and minimize the cost effort under reliability requirements.

Starting with the third section, the intuitionistic fuzzy paradigm is considered. Intuitionistic fuzzy sets, operations on IFS and TIFN are described and used to model the reliability of serial and parallel systems. Generalized fuzzy numbers and the $(\alpha, \beta)$ – cuts approach were important tools in operational development.

The intuitionistic fuzzy optimization methodology described in the fourth section is appropriate when imprecision is detected during the study of the software reliability-cost aspects. The study presented here is relevant for architectures based on components. In future, the subject will be extended for service oriented architectures.
References


Henrik Madsen, Grigore Albeanu, and Florin Popenţiu-Vlădicescu


**Henrik Madsen** (hm@imm.dtu.dk) received the M.Sc. in Engineering in 1982, and the Ph.D. in Statistics in 1986, both at the Technical University of Denmark. He was appointed Professor in Statistics with a special focus on Stochastic Dynamic Systems in 1999. He is involved in a number of cooperative projects with other universities, research organizations and industrial partners. His main research interest is related to analysis and modeling of stochastic dynamics systems. This includes signal processing, time series analysis, identification, estimation, grey-box modeling, prediction, optimization and control. The applications are mostly related to Energy Systems, Informatics, Environmental Systems, Bioinformatics, Process Modelling and Finance. He has authored or co-authored more than 210 papers and technical reports, and about 10 educational textbooks. Also, he is the head of Center for High Performance Computing at DTU which was opened by the Danish Minister of Research in February 2002.

**Grigore Albeanu** (g.albeanu.mi@spiruharet.ro) received his Ph.D. in Mathematics in 1996 from University of Bucharest, Romania. During the interval 2004 - 2007 he was the head of the UNESCO IT Chair at University of Oradea, and participates as principal investigator in a NATO project. Grigore Albeanu has authored or co-authored over 100 papers and 10 educational textbooks in applied mathematics and computer science. Since 2007, he is professor of computer science at Spiru Haret University in Bucharest, invited professor at Bucharest University, Faculty of Mathematics and Computer Science, and associate researcher at DTU Informatics. His current research interests include different aspects of scientific computing including soft computing, modelling and simulation, software reliability, virtual reality techniques and E-Learning.

**Florin Popentiu-Vlădicescu** (Fl.Popentiu@city.ac.uk) received the M.Sc. in 1975, and the Ph.D. on reliability in 1984, from the University "Politehnica" of Bucharest. He has been appointed as UNESCO Chairholder both at the City University, London and University of Oradea, Romania. He has been a visiting professor at a number of renowned European universities such as: Telecom Paris, Ecole des Mines Paris, ETH - Zurich and Université Pierre et Marie Curie, Paris. Dr. Popentiu is currently visiting professor at the Technical University of Denmark, and also at the "Grandes Ecoles in Paris". He has published over 120 papers in International Journals and Conference Proceedings and is co-author of 3 books. His research interests focuses on software safety and reliability. He is working on software reliability problems and has been Co-Director of two NATO research projects.