International Journal of Performability Engineering, Vol. 6, No. 4, July 2010, pp. 389-399.

Printed in India

Multi-objective Offshore Safety System Design Optimization

L. M. BARTLETT* and J. RIAUKE

Department of Aeronautical and Automotive Engineering, Loughborough University, Leicestershire, UK, LE11 2TU

(Received on July 09, 2009 and revised on January 25, 2010)

Abstract: The objective of this paper is to present a multi-objective approach to the design optimization process applied to systems that require a high likelihood of functioning on demand. In the real world it is common that there are several objectives to be met, not just maximising the system availability, and hence an approach is required to deal with these issues. A method is presented that integrates the latest advantages of the fault tree analysis technique and the binary decision diagram method to model the availability issue, along with a multi-objective optimization approach (the Improved Strength Pareto Evolutionary Approach) to cater for meeting the multiple criteria of assessment. The end product is a mechanism to yield the best design option. The paper presents the principles of the method and a case study to illustrate how the method is applied, along with the results produced. The case study relates to a high integrity protection system of an offshore platform. The optimization criteria involves unavailability, cost, spurious trip frequency and maintenance down time. Several enhancements to the optimization strategy to improve the efficiency of the approach are discussed.

Keywords: *safety systems, unavailability, optimization, genetic algorithms.*

1. Introduction

While designing a safety system it is necessary to maximise its availability, as its function is to work on demand. Using traditional techniques of design, test and redesign often leads to a system that is adequate in terms of performance, meeting the required safety standard, but one that is not necessarily optimal. In the real world, there is often more than one objective, and as such a multi-objective approach is required to find an optimal solution.

Much of the latest research on safety system design optimization focuses on using modern techniques for optimization, namely applying evolutionary methods. Primarily this is due to the inability of classical optimization techniques to deal with the types of objective function and constraints of the safety system design problem. In addition it is the ability of the modern heuristic optimization techniques to cater for integer variable design parameters, small search space regions, and linear and nonlinear objective function characteristics. Although system safety is of utmost importance there are often other competing objectives hence a multi–objective optimisation approach [1] is required. Nowadays one of the most powerful optimization method groups is genetic algorithms (GAs) [2]. The multi-objective GA often takes less time to find the optimal solution than other multi-objective approaches and also requires less computer resources [3]. This advantage is extremely beneficial when analysis of large safety systems is undertaken.

*Corresponding author's email address: L.M.Jackson@lboro.ac.uk

Numerous studies have shown the applicability of multi-objective GA techniques to the safety system optimization problem. The coupling of genetic algorithms and Monte Carlo simulation have been used on nuclear safety systems [4], the use of The Improved Strength Pareto Evolutionary Approach (SPEA2) based genetic algorithm approach has been used for a multiple-optimization problem, where the parameters of design, testing and maintenance act as the design considerations [5], and the method has been applied to a safety critical air traffic control system for warning about conflict between two aircraft [6]. This research and others have shown the capability of the multi-objective approach and is the focus of this paper.

For each safety system problem the specifics of the approach need to be tailored to the characteristics of the system and the constraints under analysis. This paper considers an application to an offshore safety system, which has ten design variables, four objectives being minimizing system cost, unavailability, spurious trip frequency and maintenance down time. The technique developed integrates the fault tree, binary decision diagram [7] and SPEA2 multi-objective evolutionary method [8]. Numerous enhancements to the basic methodology have been implemented to improve the efficiency of the technique developed for this specific application. The end result is an efficient methodology to find the best system design given the objectives and resources specified.

Notation

GA	genetic algorithm;
SPEA2	Improved strength pareto evolutionary approach;
BDD	binary decision diagram;
HIPSYS	high integrity protection system;
ESD	emergency shutdown,
Q_{sys}	system unavailability;
F_{sys}	system spurious trip frequency;
MDT	maintenance down time;
S_i	strength value for design i;
σ_{ij}	distance from string i to string j;
σ^k_i	k^{th} element in distance list for string i,
D_i	density value for design i;
Q'_{sys}	penalized system unavailability,

2. Application System and Design Objectives

2.1. The System

The research in this paper focuses on application to a safety system of a not normally manned offshore platform. The high integrity protection systems (HIPSYS) function is to prevent a high-pressure surge passing through it, with the aim to prevent an overpressure situation on processing equipment downstream. Figure 1 represents the main features of the HIPSYS [9]. Pressure in the system is monitored by pressure transmitters (PT). These are located within two separate subsystems, whose function is to close the valves in response to a high pressure.

The first level of protection is via sub-system 1. This system comprises a wing and a master valve, also there are three pressure transmitters fitted, with two emergency shutdown (ESD) valves (ESDV1 and ESDV2). The secondary level of protection is provided via sub-system 2. The secondary sub-system is completely independent in operation and its method of protection is the same as the primary protective mechanism. It has three pressure transmitters and 2 valves fitted (HIPS1 and HIPS2).

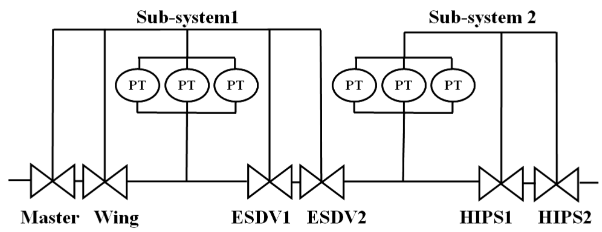

Figure 1: Structure of High-Integrity Protection System

2.2. Design Variables

To try and improve the system design various changes can be made. For this system ten design variables are defined (table 1). These changes focus on the number and type of valve fitted, the number and type of pressure transmitter fitted, also the number of transmitters required to activate the valve closure (number of transmitters to trip), and also alterations to the maintenance regime (inspection interval duration).

Table 1: Main HIPSYS Variables

Variabl e	Description	Value
θ_1, θ_2	Inspection intervals for subsystems 1 and 2	1 week – 2 years
V	Valve type	1 or 2
P	Pressure transmitter (PT) type	1 or 2
N_1, N_2	Number of PT fitted in subsystem 1 and 2 respectively	1 – 4, 0 – 4
K_1, K_2	Number of PT required to trip (activate) for subsystem 1 and 2 respectively	$1 - N_1$, $0 - N_2$
E	Number of ESD valves fitted	0 – 2
H	Number of HIPS valves fitted	0 – 2

2.3. Design Objectives

The objective of this design optimization problem is to minimize four system parameters: unavailability (Q_{sys}), spurious trip frequency (F_{sys}), cost (*Cost*) and maintenance down time (*MDT*). These parameters have been chosen as they are paramount in maintaining a high level of functionality of the system. Adding more components to system increases the cost but potentially decreases the unavailability, though depending on the type of components it may actually increase spurious trip frequency and maintenance down time impacting on increases to the unavailability. It is the balancing act between these which is paramount to ensure the best use of resources. In practice, there are three limitations (upper bounds) set on the available resources. The total cost of the system must be less than one thousand units. The average time each year that the system resides in the down state due to preventative maintenance is a maximum of 130 hours. If the number of times that a spurious system shutdown occurs is more than once per year then it is deemed unacceptable.

3. Optimization Methodology

3.1. Overview

The overall optimization method involves three embedded parts – structural representation, performance evaluation and optimization. Structural representation of the failure of the system (in terms of a fault tree) is required for all designs. Methods are required to analyze each individual design for all constraints (quantification of implicit and explicit objective functions), with a method to generate the designs to be evaluated (SPEA2 algorithm). The results from the quantitative objective analysis are then fed back into the optimization algorithm to direct the search toward the best design, as shown in Figure 2.

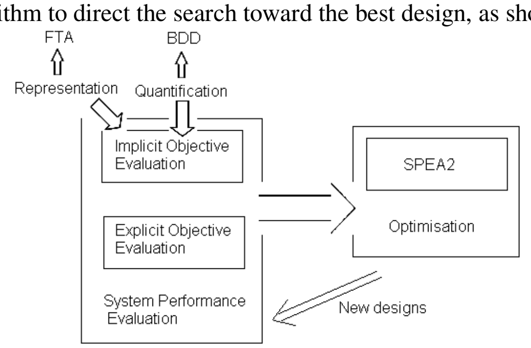

Figure 2: Optimization Methodology Summarized

3.2 System Structural Representation and Quantification

For any design optimization problem representation of the system under analysis is required. Given the system considered in this research has the need to function on demand, assessing this performance criteria is essential. A commonly used technique for failure evaluation is fault tree analysis. For the design optimization problem, house events [10] can be included in the tree to allow modelling for all design alternatives. It prevents the need for separate fault trees for each design. Branches of the fault tree can then be turned on or off depending on the specific design structure under analysis. For example, in figure 3, if a valve of type 1 is fitted then the house event HE0 is set to true and HE2 is set to false, permitting analysis of the system with valve type 1 fitted.

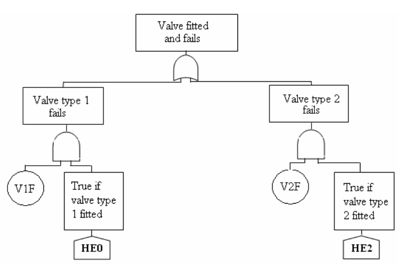

Figure 3: House events

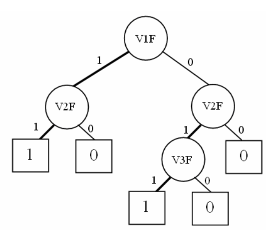

Figure 4: Example BDD

In terms of quantification of the fault tree the BDD technique [7] is used. This technique changes the form of the diagram into a form that is more easily manipulated mathematically. The BDD can be described as a rooted, directed acyclic graph (as shown in figure 4), which is comprised of nodes (representing components, *i.e.,* V1F is Valve type 1 fails) and edges (representing component states, 1 is the failed state and 0 the working state). The end terminal square boxes represent the state of the system. Analysis is computationally more efficient and can deal with large fault trees without the requirement for approximation analysis. The quantification of the objectives for a problem depends on their form. There are two main categories: explicit and implicit. Explicit ones can be determined and easily evaluated from an explicit function of the design variables. In contrast, implicit constraints can only be evaluated by a full analysis of the system. This research involves quantification of both types of objective.

3.3. Optimization Algorithm

Among all major multi-objective evolutionary algorithms, the SPEA approach [11] is one of the most popular. Advances in the method have yielded the SPEA2 architecture [8]. The suggested overall algorithm for this research is:

Step 1: *Initialization*: Generate an initial population and create the empty archive (external set).
Step 2: *Fitness assignment*: Calculate fitness values of individuals in initial population.
Step 3: *Environmental selection*: Copy all nondominated individuals to the archive. If its size exceeds the allowable size then reduce the archive by means of the truncation operator, otherwise fill the archive with dominated individuals from initial population. Note: the archive size is constant over time.
Step 4: *Termination*: If the maximum number of generations is reached then set the nondominated set to the set of decision vectors represented by the nondominated individuals in the archive. Stop.
Step 5: *Mating selection*: Perform binary tournament selection with replacement on the archive in order to fill the mating pool.
Step 6: *Variation*: Apply recombination and mutation operators to the mating pool and set the archive to the resulting population. Increment generation counter and go to *Step2*.

4. Implementation of Optimization Methodology to HIPSYS

4.1 HIPSYS structural representation

The C++ package was used to build the optimization methodology software. The first part of the program produces the HIPSYS design architecture. A full system analysis is required for the evaluation of the system unavailability. The top event of the HIPSYS unavailability fault tree represents the causes of the system failing to protect the processing equipment. In total the fault tree consists of 154 gates, 38 basic events representing component failures, and 40 house events representing the design options.

The spurious trip frequency for each design is also an implicit objective that requires the use of fault tree analysis to assess its value. The causal relationship ‘HIPSYS fails spuriously’ is represented by the sub-events ‘Wing or Master Valve Fails Spuriously’, ‘ESD Subsystem Fails Spuriously’ and ‘HIPSYS Subsystem Fails Spuriously’ related by ‘OR’ logic. The fault tree consists of 142 gates, 38 basic events and 40 house events.

For each design generated by the optimization approach, the process of calculating the unavailability and spurious trip frequency for each specific design has three stages. Each new design is checked for feasibility (stage 1). House events are set in the fault tree structure to represent the specific design (stage 2). The fault tree is then reduced to remove redundant branches (stage 3) in preparation for analysis via the BDD method in the quantification phase.

4.2. Quantification of Objectives

This stage involves evaluation of the objectives. For the fault tree unavailability and the spurious trip frequency calculation each HIPSYS design undergoes a BDD construction then evaluation stage. Standard BDD quantification methods are used.

The remaining objectives require mathematical calculation. Cost of the HIPSYS design can be calculated using equation 1.

$$Cost = Cost(subsystem1)+Cost(subsystem2) \leq 1000 \quad (1)$$

The cost of each sub-system depends on the number of valves fitted as well as the cost of the valves of type 1 and type 2, the cost of the PT of type 1 and 2. There is also a constant included to accommodate the fixed costs of both subsystems.

Similarly, the average maintenance down time (MDT) is calculated as a sum of the maintenance down time of subsystem 1 and subsystem 2 for each potential design (equation 2):

$$MDT = MDT(subsystem1) + MDT(subsystem2) < 130 \quad (2)$$

Included are the number of valves, the test times of the valves of type 1 and type 2, and the test times of the pressure transmitter of type 1 and 2. Again there are constants referring to the sum of the test times for the fixed components in each subsystem. Full details are given in reference [9].

Limitations are set on three of these objectives and penalties are incurred on the unavailability value when violation occurs (these are explained in detail in reference 12). The resulting value is a penalized system unavailability, which participates in the optimization procedure. If no violation occurs the initial unavailability value is used.

4.3 Optimization Implementation

4.3.1 Coding and Initialization

The number of strings for the initial population was set at 20. These are generated randomly. Each design variable must be allocated a particular length of the string *i.e.,* a particular number of bits, in order to accommodate the largest possible value in binary form. In total, each string is 32 bits in length, as shown in Figure 5.

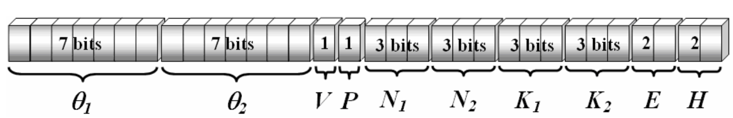

Figure 5: Binary Representation of Solution String

4.3.2 Genetic Algorithm Fitness Assignment

The quality of potential designs are calculated by creating a fitness score relating to dominance. Since this is a minimization problem, design *a* is said to dominate design *b* if all *a* parameter values are equal to or smaller than parameter *b* values or at least one of parameter *a* values is smaller than the respective *b* parameter value. Initially each string is assigned a strength value $S(i)$, representing the number of solutions it dominates.

On the basis of the *S* values, the raw fitness $R(i)$ of a design *i* is calculated. This fitness is determined by summing the strengths of its dominators in both the archive and population (*i.e.,* those designs with higher strength values). As there is the potential for most designs to not dominate each other, additional information is incorporated to discriminate between designs having identical raw fitness values. The density estimation technique used in SPEA2 is an adaptation of the k^{th} nearest neighbour method [8], where the density at any point is a decreasing function of the distance to the k^{th} nearest data point. In this problem the inverse of the distance to the k^{th} nearest neighbour is taken as a density estimate σ_{ij} , *i.e.,* for each individual i the distances to all designs j in the archive and population are calculated using equation 3:

$$\sigma_{ij} = \sqrt{(C(i) - C(j))^2 + (MDT(i) - MDT(j))^2 + (Q'(i) - Q'(j))^2 + (F_{sys}(i) - F_{sys}(j))^2} \tag{3}$$

where $C(i)$ and MDT(i) are the cost and maintenance down time of the i^{th} design respectively, $Q^{'}(i)$ and Fsys(i) are the i^{th} designs penalized system unavailability and spurious trip frequency respectively, j is from the interval [1,.., population size] with the condition that $i \neq j$. Obtained distances are stored in a list (matrix). After sorting the list in increasing order, the k^{th} element gives the distance sought for design i, denoted as σ_i^k , where k is equal to the square root of the population size. Afterwards, the density $D(i)$ corresponding to i is defined by equation 4.

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{4}$$

In the denominator, two is added to ensure that its value is greater that zero. The fitness is calculated by adding the two factors, the raw fitness and the density function.

5. Case Study Results

5.1 Optimization Schemes

Two different optimization schemes have been implemented to tailor the algorithm parameters for the HIPSYS system in order to evaluate the one that leads faster to the global optimal solution. In the first scheme a single population of 20 strings have been generated and run through 3000 generations with the crossover and mutation rates equal to 0.7 and 0.01 respectively. The second scheme was based on thirty different initial populations with only 100 generations for each run of the optimization program with the same crossover and mutation rates. The first scheme resulted in a single Pareto set of nondominated HIPSYS design options, on the other hand the second scheme gave 30 sets. A Pareto set obtained from the 20^{th} run in the second scheme consisted of a larger number of nondominated solutions by most optimization parameter values and, therefore, has been chosen for comparison with the first optimization scheme.

All results showed that both optimization schemes produced very similar Pareto fronts, however the front obtained by the first scheme produces a larger number of the boundary points (design solutions) due to a larger number of generations. Observation of the data itself shows that the 1^{st} scheme produces up to 4 additional non-dominated designs.

The comparison of the best results obtained during experiments is shown in Table 2. The minimal unavailability values concentrate in the intervals [4.235e-7, 1e-6) and [4.143e-7, 6e-7) for the first and second schemes respectively. There are also slight differences in the values of the other objectives. The parameter values are also very similar for all designs. The obtained results prove the good performance of the developed tool, which produces a solution close to the global minimum in only 20 minutes (for 3000 system evaluations). Despite the relatively small number of generations, the second optimization scheme provided larger diversity between potential HIPSYS designs and lead to a smaller system unavailability. These results compared well with those from using an exhaustive search method with unavailability as the objective function. The slight differences were due to the competing nature of the four objectives in the multi-objective algorithm.

Table 2: Results Comparison

Parameters	Scheme 1	Scheme 2	Parameters	Scheme 1	Scheme 2
No. of ESD valves, E	0	0	Mainten int, θ_2	60	93
No. of PTs, N_1	2	3	Valve type V	1	1
No. of PTs to trip, K_1	2	2	PT type	1	1
Maintenance int, θ_1	19	18	MDT	128.40	129.53
No. of HIPS valves, H	1	1	Cost	632	652
No. of PTs, N_2	2	3	Fsys	0.45044	0.45027
No. of PTs to trip, K_2	2	2	Qsys	4.235e-7	4.14e-7

5.2 Enhancements to the Algorithm

5.2.1. Overview

Though the results from the algorithm produce a pareto optimal set there are modifications to some of the optimization parameter values and some parts of the program that could possibly improve the method further. These modifications relate to: the optimization parameter values (population size, crossover rate, mutation rate); the crossover procedure (not single-point), and the methods for changing infeasible parameter parts to feasible ones (regeneration). The exact tuning and end results of the changes made are problem specific however the basis of the modifications used can be applied in other problem domains.

5.2.2 Crossover and Mutation Rates Modification

According to research [2] into the effectiveness of the GA operators, the optimal values for the crossover rate appear to be in the range from 0.5 to 0.9. The interval for the effective mutation rate values is (0, 0.2). Hence, investigation was produced with a limited set of values for each parameter. The crossover rates chosen were 0.5, 0.6, 0.7, 0.8 and 0.9. Mutation rates were 0.1, 0.01 and 0.001.

In total there are 15 possible combinations of these parameters. Hence, 15 runs of the optimization program were carried out. There were three optimal solutions among the obtained possible system designs with parameter values which are the best according to the considered limitations. The unavailability of the first design is the smallest of all obtained solutions (Q_{sys} = 4.504e-7), the second design gives the smallest cost (512 units) and the third design benefits from the smallest spurious system failure (F_{sys} = 0.245132) and MDT closest to the limit of 130 hours (MDT = 129.9834). There is not one combination of parameters which yields the smallest values for all objectives, hence the best performance result should be selected given the optimization objectives given more weighting. In this case unavailability is chosen given the characteristics of the system under analysis and hence the first solution is assumed the best for this application (crossover rate of 0.7 and mutation rate of 0.01).

5.2.3 Modification of the Population Size

The populations of 5, 10, 20 and 40 strings have been investigated by running the program with 0.7 crossover rate and 0.01 mutation rate, given the results following the crossover and mutation rate investigations. As it might be expected, the larger population lead to a better performance as each time when the population size doubles, the average values of MDT, system spurious failure and unavailability are improved. However, the best design parameters values change chaotically. Similar to the earlier results, all optimal designs are very close to each other. However, with more objectives being lower and if using unavailability as our primary concern, the population of 20 strings produces the best results for this application.

5.2.4. Modification of the Parameter Evaluation Scheme

There are four HIPSYS design parameters (number of pressure transmitters fitted and required for subsystems 1 and 2, *i.e.*, N_1, N_2, K_1 and K_2) in which their conversion from binary form can yield infeasible values. The initial approach was to rigidly assign infeasible values to alternatives depending on the decimal value. For the modified parameter evaluation scheme if the parameter value is infeasible, this parameter is regenerated. The process stops only when the new value is from the feasible region. The average results, produced by the modified method compared to the initial approach were better in terms of three optimization parameter values (cost, MDT and unavailability) hence enhancing the optimization algorithm.

5.2.5 Modification of the Crossover Procedure

During the investigation process three additional crossover procedures were created. The modified method is similar to the single-point crossover. The main difference is that the second parent string from the pair can again participate in crossover as the first parent. Two-point and Three-point crossover methods first generate respectively two and three random positions of the string. Then parent strings are crossed at those points. All methods were applied to the different sized populations (5, 10, 20 and 40 strings).

As it was expected the new crossover methods produced poor results for the smallest population since it is not diverse. The obtained results were quite chaotic with the minimal objectives being scattered across crossover methods. For all three populations (10, 20 and 40 strings) the modified crossover method worked best since it produced the largest number of the best parameter values and hence was deemed best suited to this application.

5.3 The Chosen Optimization Scheme

According to the investigation results the following modified optimization scheme was chosen:

- Population of 20 strings;
- 0.7 crossover rate;
- 0.01 mutation rate;
- 100 generations;
- Modified parameter estimation procedure;
- Modified crossover operator.

Tables 3 and 4 show the comparison of the best design parameter values obtained by the initial and modified programs. All solutions are nondominated. The initial version of the program gives smaller unavailability for both the best and average results. However, the modified algorithm produces slightly better values for the system cost, MDT and spurious system failure. The average cost was reduced by 27 units. The average MDT was improved and is reasonably closer to the limit of 130 hours. The spurious system failure of the new best design is two times smaller. The difference between the initial and modified unavailability is 0.00005. If this difference is insignificant for the potential decision maker then the modified program results can be assumed to be better. Throughout the process as multiple solutions are produced the analyst must make a choice as to which objective gains precedence. Ultimately this research has shown the applicability of the method to such systems.

Table 3: Best Design Characteristics

Pop size	Estimation technique	Parameter Values	Cost	MDT	F_{sys}	Q_{sys}
20 strings	Initial	Average	597	120.718	0.37532	6.9379e-5
		Best	592	129.701	0.45468	4.5042e-7
	Modified	Average	570	126.702	0.32278	5.0000e-4
		Best	592	129.983	0.24513	5.0500e-5

Table 4: Best Designs Comparison

Best Design	Q1	Q2	E	H	N1 / K1	N2 / K2	V	P
Initial	25	73	0	1	1 / 1	3 / 3	1	2
Modified	29	108	1	0	2 / 2	0 / 0	1	1

6. Conclusions

An integrated optimization approach has been developed and discussed in this paper. The technique has been successfully applied to the high integrity protection system optimization problem. Results show that this technique produces a set of non-dominated solutions which yield optimal designs given the four objectives and resource limitations. The parameters of the optimization approach and alterations to the design parameter allocation methods show the flexibility of the technique for this application. The overall conclusion is that this method is suitable for such safety system design problems and the authors can see that the method has potential for application to systems in other domains. Further work would be beneficial to examine scalability and complexity issues.

References

[1]. K. Deb. Multi-objective optimization using evolutionary algorithms. John Wiley and Sons, Chichester, 2008.

[2]. D. E. Goldberg. *Genetic Algorithms in Search. Optimization and Machine Learning*, Addison-Wesley Publishing Company, Boston, USA, 1989.

[3]. I. F. Sbalzarini, S. Muller, and P. Koumoutsakos. *Multiobjective optimization using evolutionary algorithms*, Center for Turbulence Research, Proceedings of the Summer program 2000, Stanford, USA, pp. 63-74, 2000.

[4]. M. Marseguerra, E. Zio, and L. Podofillini. *A Multiobjective Genetic Algorithm Approach to the Optimization of the Technical Specifications of a nuclear safety system*, Reliability Engineering and System Safety 84, 1, pp. 87-99, 2004.

[5]. S. Martorell, A. Sanchez, S. Carlos, and V. Serradell, *Alternatives and Challenges in Optimizing Industrial Safety Using Genetic Algorithms*, Reliability Engineering and System Safety 86, 1, pp. 25-38, 2004.

[6]. R. M. Everson, and J. E. Fieldsend. *Multiobjective Optimization of Safety Related Systems: An Application to Short-Term conflict Alert*, IEEE Transactions on Evolutionary Computation, University of Exeter, UK, Vol. 10, 1, pp. 187-198, 2006.

[7]. A. Rauzy. *New Algorithm for Fault Tree Analysis*, Reliability Engineering and System Safety, Vol. 40, 3, pp. 203-211, 1993.

[8]. E. Zitzler, M. Laumanns, and L. Thiele. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, Computer Engineering and Communication Network Lab (TIK), Swiss Federal Institute of Technology, TIK-Report No. 103, 2001.

[9]. J. D. Andrews, and R. L. Pattison. *Genetic Algorithms in Optimal Safety System Design*, Proc. Instn. Mech. Engrs., Vol. 213, 3, pp. 187-197, 1999.

[10]. J. D. Andrews, and T.R. Moss. *Reliability and Risk Assessment*. Second Edition. Professional Engineering Publishing, London, 2002.

[11]. E. Zitzler, and L. Thiele. *An Evolutionary Algorithm for multiobjective Optimization: The Strength Pareto Approach*, Computer Engineering and Communication Network Lab (TIK), Swiss Federal Institute of Technology, TIK-Report No. 43, 1998.

[12]. J. Borisevic, and L. M. Bartlett. *Safety System Optimization by Improved Strength Pareto Evolutionary Algorithm (SPEA2)*, Proceedings of the 17th AR^{2}TS, pp. 38-49, 2007.

J. M. Bartlett (*For her biography, please see page 190 of IJPE, Vol.6, No. 2, March 2010)*