An integrated design optimisation approach for systems with dependencies

An Integrated Design Optimisation Approach for Systems with Dependencies

J.Riauke[1] and L.Bartlett[2]
[1] Advantica a GL company, Loughborough, Leicestershire, UK
[2]Aeronautical and Automotive Engineering , Loughborough University, Loughborough, UK

Abstract

The design of a safety system is critical if functionality is to be maximised and consequences reduced. There is often a trade off between the performance obtainable and the resources available. To address these balancing issues, which are usually impractical by hand for a designer, multi-objective optimisation techniques can be used. When considering safety systems there is often the situation of dependencies between components, for example with regard to maintenance. To evaluate the system behaviour in these situations an appropriate analysis method is required. The aim of this paper is to present an optimisation approach which integrates traditional methods of system failure evaluation. The combined method uses the fault tree analysis technique to represent the causes of failure on demand of the system, the binary decision diagram and Markov methods for system quantification (for independent and dependent sections of the fault tree respectively), and the Improved Strength Pareto Evolutionary Approach (SPEA2) to find the most optimal design solution. The end product is a mechanism to yield the best design option for safety systems incorporating dependencies. The paper presents the principles of the method and a case study to illustrate how the method is applied. The results produced, along with conclusions are provided.

1. Introduction

When designing a system using the traditional techniques of design, test and redesign the output is often adequate in terms of performance, meeting the required safety standard, but one that is not necessarily optimal. Attempting to optimize the design of engineered safety systems, the analyst is frequently faced with the demand of achieving several targets (e.g. low costs, high revenues, high reliability, low accident risks), some of which are often in conflict. At the same time, several requirements (e.g. maximum allowable cost, weight, volume etc.) should also be satisfied. Traditionally this type of problem has been solved by focusing the optimisation on a single objective which may be a weighed combination of some of the targets of the design problem. More recently the benefits of applying a multi-objective approach have been identified.

During the last decade a number of engineers have applied various methods for safety system optimisations. Preference has swayed to using modern approaches, evolutionary methods, due to their ability to cater for integer variable design parameters, small search space regions, and linear and nonlinear objective function characteristics. Among these methods genetic

algorithms have been applied most often due to their simplicity and universality, with considerable success. Cantoni et al. [1], Busacca et al. [2], Marseguerra et al. [3], Martorell et al. [4], and Everson and Fieldsend, [5], all have applied a multi-objective optimisation procedure incorporating the genetic algorithm principle to safety related and critical systems producing optimal outcomes. 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 research and others have shown the capability of the multi-objective approach and is the focus of this paper.

To extend the applicability of using an optimisation approach there is a need to deal with systems that have dependent characteristics. The method of optimisation developed in this research integrates techniques that can accommodate such dependencies. This paper considers an application to an offshore safety system (a high integrity protection system), which has ten design variables, four objectives, and maintenance type dependencies. The objectives relate to minimizing system cost, unavailability, spurious trip frequency and maintenance down time. The dependencies exist between groups of components which are maintained by the same engineer. The approach developed integrates the fault tree, binary decision diagram, Markov and multi-objective evolutionary methods. Dependencies within the system are highlighted within the fault tree structure and these sections are analysed with the Markov approach. This keeps the analysis of such sections to a minimal form and hence aids the efficiency of the overall optimisation approach. The research uses the multi-objective improved strength Pareto evolutionary approach (SPEA2) [6]. The results of the research illustrate the applicability of the approach and the need to consider dependencies to prevent an underestimation of the system performance.

The remainder of this paper is divided into four sections. The second section considers the integrated optimisation method. Section 3 overviews the application safety system and defines the optimisation objectives, with the fourth section detailing the implementation to the case study problem. Section 5 discusses the results obtained, with the main conclusions given in the final section.

2. Integrated Optimisation Method

2.1 Overview

Previous research [7] has illustrated the suitability of combining the well-known techniques of fault tree analysis and the binary decision diagram method within an optimisation approach. The use of these methods is adequate only if all component failures occur independently, since the techniques are not able to take into account the dependencies. It is rare that real safety systems consist of just independent components in terms of failure or (and) repair. Failure to identify the dependency in the system would result in an incorrect system performance evaluation (for example, unavailability and failure frequency prediction). Therefore, an appropriate modelling technique is required to overcome the problems. This papers introduces a new

methodology enabling optimisation of safety systems with dependencies by effective use of the Markov modelling tool. Incorporation of the standard fault tree analysis (FTA) and binary decision diagram (BDD) methods are maintained. An overview of the approach is given in figure 1 and discussed in sections 2.2-2.4.

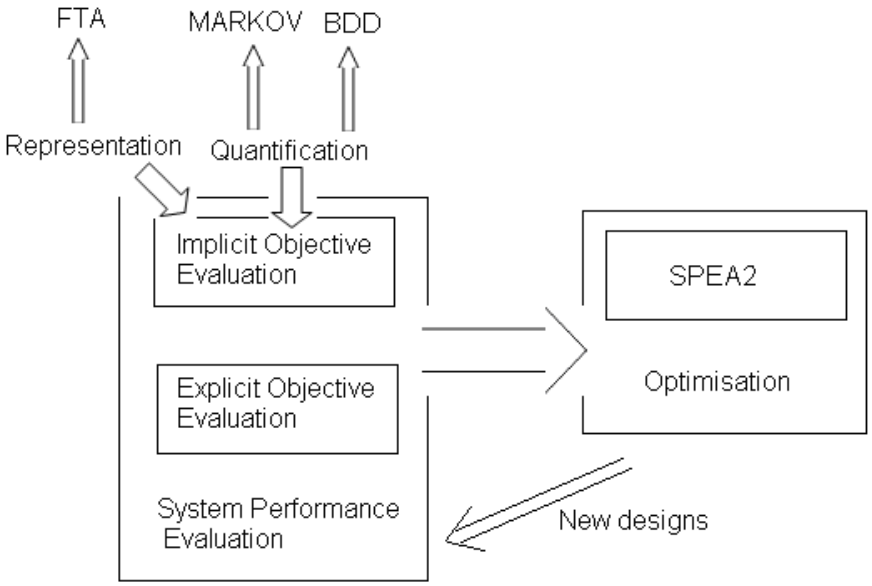

Figure 1 – Flow chart of optimisation approach

2.2 Explicit Objective Evaluation

There are two main categories of objectives: explicit and implicit. Explicit ones can be determined and easily evaluated from an explicit function of the design variables. In contrast, implicit objectives can only be evaluated by a full analysis of the system. Cost is an example of an explicit objective, where a function can be generated for evaluation. The form of each function is specific to each optimisation application.

2.3 Implicit Objective Evaluation

2.3.1 Representation: As the application of the developed optimisation methodology is to safety systems, the primary performance criteria relates to its functionality. Changing design variables will change the system functionality representation therefore a full analysis is required for each design option. System performance measures can be obtained by using the fault tree analysis method. Fault trees are used to quantify the system unavailability of each potential design. Constructing a fault tree for each design variation would be a time consuming task, hence, impractical. To overcome this house events [8] can be used. These enable the construction of a single fault tree capable of representing the causes of the system failure mode for every possible system design.

2.3.2 Quantification: To analyse the fault tree used for representation of the performance of the system the latest BDD technique has been used. The If-Then-Else construction method developed by Rauzy [9] is implemented. The method allows exact system quantification in a more efficient manner than the traditional kinetic tree theory technique.

The additional feature of the optimisation approach developed is the inclusion of the Markov method to evaluate sections of the fault tree where there are dependent events. The process involves the following steps[10]:

> **Step 1** - Fault tree simplification and modularization in order to obtain independent modules containing specific dependency groups.
> **Step 2** - Markov analysis of these groups.

As a result, the conventional fault tree structure is maintained. To illustrate consider the fault tree given in figure 2, where the top event (*Gate1*) has two dependency groups: group one is represented by events A and B (coloured in grey), and group two consists of events D and E (identified with a striped pattern).

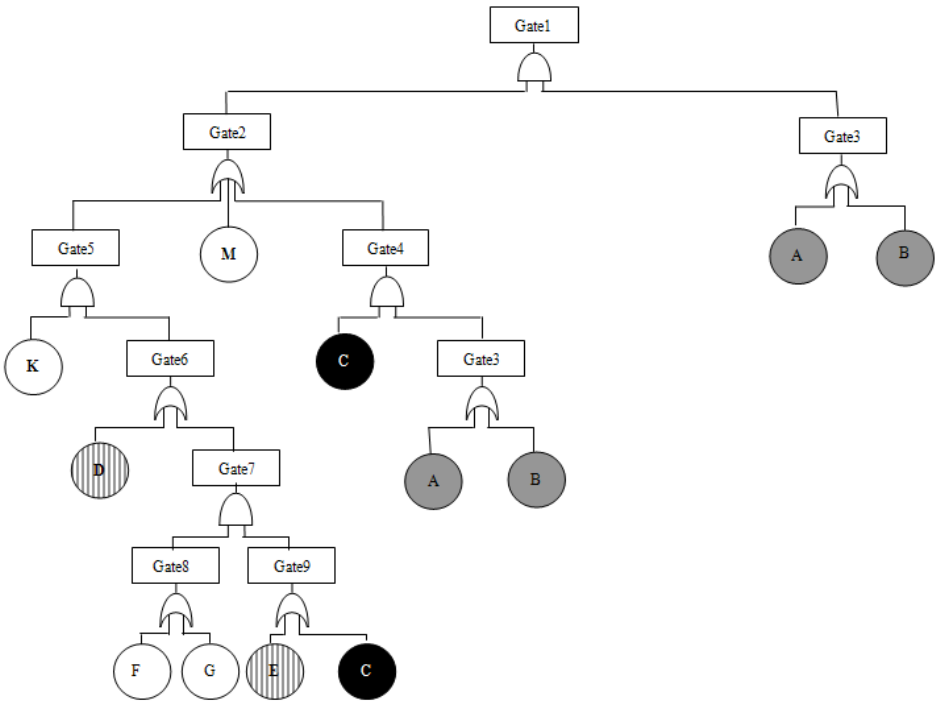

Figure 2 – Example fault tree including dependencies

During the modularisation process in finding modules of the tree some are relatively straightforward, for example Gate3. Its inputs always occur together with no other elements from the tree and hence form a module. However other elements make identifying modules more difficult. For example, when trying to group parts of the fault tree containing the dependency group of D and E the event C (coloured in black) is an obstacle as it occurs elsewhere in the tree. In such circumstances modules within modules are created. This enables the smallest Markov model to be constructed at all times. The end result for this example fault tree is given in figure 3, where Mod 1 represents

Gate3. It is a minimal module for the dependency group with A and B. Module 2 (Mod2) is an OR combination of Gate5 and Gate4. It should be noted that Gate4 includes Gate3, which will be replaced by module 1 (Mod1).

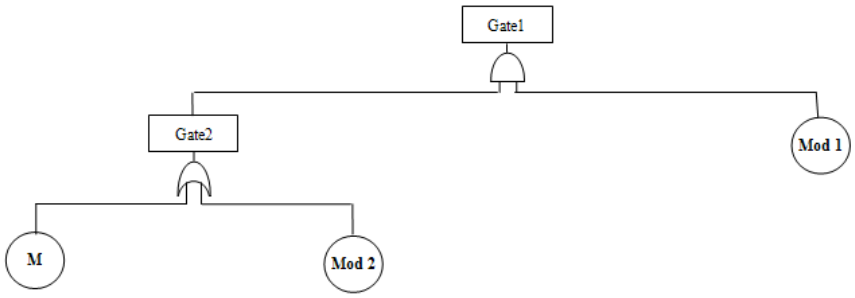

Figure 3 – Resulting modularised fault tree

When the fault tree modularization into independent sections is finished, the analysis of these independent sub-trees will be carried out by the Markov method, if the section contains dependent components; or by the binary decision diagram, if all sub-tree components are independent. For instances where there are multiple modules in dependent sections of the tree, each are analysed in turn. For the example in figure 3, when the Markov analysis starts, Mod 1 will be analysed first and changed to a basic event and, hence, enable the analysis of Mod 2. If the dependent component groups could not be separated even after the use of the suggested technique, the whole fault tree is treated as a module and goes through the Markov analysis.

2.4 Optimisation Methodology

The main goal of multi-objective optimisation is the search for acceptable solutions to problems that incorporate several performance criteria. The technique used to find the optimal system design incorporates the issues of pareto optimality and dominance, and is the Improved Strength Pareto Optimisation Approach (SPEA2) [6]. The algorithm works through six steps, as follows:

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. Important notice: the number of individuals contained in the archive is constant over time.
Step 4. *Termination*: If the maximum number of generations is reached or another stopping criterion is satisfied 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*.

3. Application Safety System

3.1 The system

A safety system of a not normally manned offshore platform is considered in this research. The high integrity protection systems (HIPS) function is to prevent a high-pressure surge passing through it, with the aim to prevent an overpressure situation on processing equipment downstream. Figure 4 represents the main features of the HIPS [12].

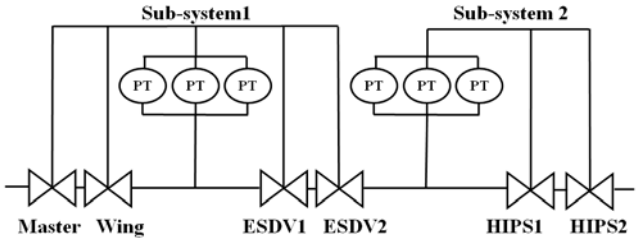

Figure 4 - Structure of High-Integrity Protection System

There are two separate subsystems involved, the first level of protection is via the Emergency Shutdown sub-system (sub-system 1) and a secondary level of protection is via the high integrity protection system (sub-system 2). The first level of protection acts to close the Wing and Master valves together with any valves that have been fitted when pressure in the pipeline exceeds the permitted value. This value is monitored using pressure transmitters (PT). The secondary sub-system is completely independent in operation and its method of protection is the same as the primary protective mechanism.

3.2 Optimisation Objectives

The objective of this design optimisation 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 influencing factors in maintaining a high level of functionality of a system required to operate on demand. As a multi-objective approach is used there is a balance between the four objectives, and in practice the choice of system design is not unlimited. In this case, there are three limitations (upper bounds) set. 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 one hundred and thirty hours. If the number of times that a spurious system shutdown occurs is more than once per year then it is deemed unacceptable.

3.3 Design variables and limitations

For many design problems given only a relatively small number of design changes the list of potential designs quickly becomes impractical to evaluate by hand. For this system ten design variables are defined (table 1). One assumption made in this analysis is that when a valve type is selected, all valves are fitted as this type. For each component the information required for the analysis includes failure data, maintenance times and costs. For the failure data each component can fail either in a dormant mode or spuriously. A dormant failure can be described as the inability of the component to carry out its desired task on demand. In contrast, spurious failure results from the component carrying out its desired function when its operation is not required. This data will be used subsequently when calculating the unavailability and spurious trip probability of the HIPS. The cost and maintenance data is used for calculation of the remaining two objectives.

Variable	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 type	1 or 2
N_1, N_2	Number of pressure transmitters fitted in subsystem 1 and 2 respectively	1 – 4, 0 – 4
K_1, K_2	Number of pressure transmitters 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

Table 1 - Main HIPS variables

The main types of dependency which are frequently encountered in many safety systems are maintenance, standby, secondary failure, initiator-enabler and test dependencies [8]. The focus of this paper is to cater for the issue of maintenance dependency (other dependency situations are discussed in the conclusions), which is common for all safety systems. This situation arises when one maintenance engineer or a team of engineers has to take responsibility for a group of components usually of the same or similar type. If several components from the same maintenance group fail subsequently, only one of them goes through the repair process. Others wait in a queue for repair until the engineer has restored the first component. The queuing affects repair times and, hence, the probability of failure of other components. Therefore, the maintenance dependency affects the whole system and influences its performance statistics. In total eight dependency groups have been identified for the HIPS structure. Group one links the pressure transmitters of type 1, group 2 the pressure transmitters of type 2, group 3 includes subsystem 1 elements - wing valve, master valve, ESD valves (type 1) and also subsystem 2 HIPS valves of type 1. Group 4 is the same as group 3 but for type 2 components. These four groups relate to the unavailability fault tree and similar groups are constructed for the spurious trip fault tree. It is assumed that each group is maintained by one engineer.

4. Case Study Implementation

4.1. Explicit Objectives

The explicit objectives relate to the derivation of system cost and maintenance down time. Cost of the HIPS design can be calculated using equation 1.

$$Cost = Cost(subsys1)+Cost(subsys2) \leq 1000 \quad (1)$$

The cost of each sub-system includes the cost of the valves of type 1 and type 2, the cost of the PT of type 1 and 2, and the cost of the solenoid valves. 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):

$$MDT = MDT(subsys1) + MDT(subsys2) < 130 \quad (2)$$

Included in this formula are the test times of the valve of type 1 and type 2, the test times of the pressure transmitter of type 1 and 2, and the test time of the solenoid valve. 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 [7].

Limitations are set on 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 optimisation procedure.

4.2 Implicit Objectives

4.2.1 Representation: A full system analysis is required for the evaluation of the system unavailability. The top event of the HIPS unavailability fault tree represents the causes of the system failing to protect the processing equipment. The top event ‘Safety system fails to protect’ will occur if all (Wing, Master, ESD and HIPS) valves along the pipeline fail to close. 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. House events are again used to construct a fault tree capable of representing each potential design for this failure mode. The causal relationship ‘HIPS fails spuriously’ is represented by the sub-events ‘Wing or Master Valve Fails Spuriously’, ‘ESD Subsystem Fails Spuriously’ and ‘HIPS Subsystem Fails Spuriously’ related by ‘OR’ logic. The fault tree consists of 142 gates, 38 basic events and 40 house events.

4.2.2 Quantification: The C++ package was used to build the HIPS optimisation software called ISPEASSOP (Improved Strength Pareto Evolutionary Algorithm Safety System Optimisation Procedure). There are three main parts of the program. Part one is responsible for the HIPS structure, part two is responsible for quantitative analysis and part three is the implemented SPEA2 algorithm for the HIPS optimisation. The key steps for the quantification phase are summarised in figure 5.

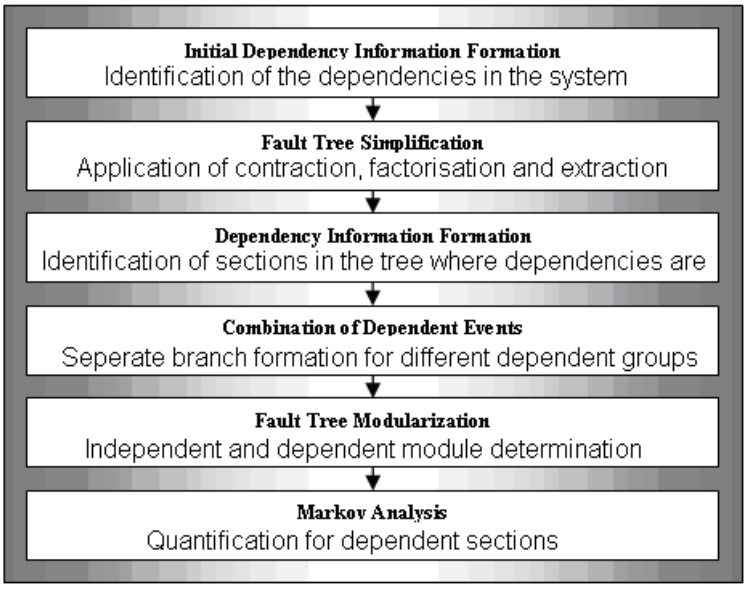

Figure 5 - Program structure for quantification phase

These steps are repeated for each design created during the optimisation process. After quantification of the dependent sections these are submitted back into the fault tree and analysis is carried out using the standard BDD technique. The outcome is a value for unavailability or spurious trip frequency (depending on the tree analysed) which is used within the optimisation process to determine the 'best' design option.

4.3 Optimisation Parameters

To implement the SPEA2 optimisation methodology some predefined values are required. After testing, the values which yielded the most efficient approach were: a population (and archive) of 20 strings; a probability of 0.7 for the crossover rate; a probability of 0.01 for the mutation rate; and a termination criterion of 100 generations.

In terms of the representation within the algorithm, each design option forms a string where a binary coding was used. Each parameter was 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 representing all design variables is 32 bits in length. It can be interpreted as a set of concatenated integers in binary form, as shown in figure 6.

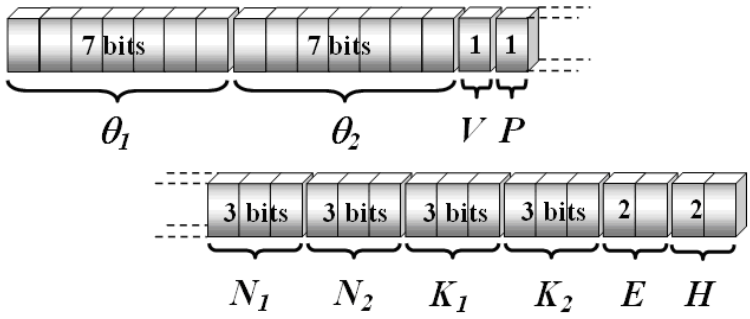

Figure 6 - Binary representation of solution string

A modified crossover operator was used, which was a deviant of the single point crossover process. The procedure works through the following steps:

Step 1. The random number is generated.
Step 2. If the generated number is smaller than the crossover rate, the pair of population strings *j* and *j+1* are crossed at the randomly chosen position. If not, step one repeats for the pair of strings *j+1* and *j+2*. (This would be strings *j*+2 and *j*+3 in the traditional single point crossover process).
Step 3. If population end is not reached the process repeats from step one for the next string in the population.

5. Results

The results from the optimisation of the HIPS with dependencies have been tested against those obtained for the same system ignoring any dependencies (in this case the standard FTA and BDD quantification occurs). The best ten designs obtained after ten runs of each program have been compared. Table 2 shows the results (WO refers to without dependencies, W with dependencies) . The corresponding design parameter values are provided in table 3.

Run No.	Cost		MDT		F_{sys}		Q_{sys}	
	WO	W	WO	W	WO	W	WO	W
1	592		129.7008		0.455	0.476	4.50e-7	5.32e-7
2	512		129.6974		0.332	0.389	8.33e-4	9.26e-4
3	582		128.7361		0.324	0.350	6.80e-4	7.25e-4
4	922		128.2273		0.718	0.766	1.00e-6	8.20e-6
5	882		129.1590		0.166	0.235	1.00e-6	1.60e-6
6	992		129.2523		0.552	0.612	1.00e-6	8.04e-6
7	852		128.3286		0.245	0.295	6.55e-4	1.06e-3
8	542		128.9881		0.324	0.387	8.45e-4	9.01e-4
9	872		129.9032		0.377	0.437	1.00e-6	5.93e-6
10	862		129.7309		0.999	0.999	1.00e-6	8.88e-6

Table 2 - Comparison of the results with and without dependencies

It can be seen from table 2 that implementation of maintenance dependency for the HIPS components resulted in a higher system unavailability and spurious trip frequency values for all designs. These changes can be explained by the increase of repair times for individual components due to the implemented dependency. In terms of selecting the ideal safety system design any one of the 10 could be chosen depending on the particular objective given the most precedence.

Run No.	*Q1*	*Q2*	*V*	*P*	*N1*	*N2*	*K1*	*K2*	*E*	*H*
1	25	73	1	2	1	3	1	3	0	1
2	27	105	2	2	1	0	1	0	1	0
3	64	9	2	1	4	0	3	0	1	0
4	33	96	1	1	2	3	1	3	1	1
5	42	53	1	2	4	2	4	1	1	1
6	34	90	2	2	2	3	2	2	1	2
7	40	91	1	2	3	0	3	0	2	0
8	27	118	2	1	2	0	2	0	1	0
9	26	124	1	2	3	2	3	2	0	2
10	42	46	1	2	2	2	1	2	1	1

Table 3 - Design parameter values for Table 2

6 Discussion

The fault tree simplification and modularization methods along with the Markov analysis have been incorporated into a new optimisation tool in order to allow the effective search for an optimal safety system design with multiple objectives and maintenance dependency issues.

The new technique has been demonstrated on the HIPS system. Comparison of results utilising the assumption that the system components are all independent and those, obtained by the suggested optimisation technique for the system with dependencies, shows that for all potential original HIPS designs the system unavailability and spurious failure frequency have been underestimated. Therefore, it is important to identify all system dependencies for more accurate system unavailability and spurious failure frequency prediction.

In its current form the developed methodology widens the field of application for optimisation studies. Consideration of other dependency types could easily be added to the programs capabilities to further enhance its scope. The main weakness of the optimisation tool is its running time. One run of the program for independent system components is in order of minutes, however, the dependent version requires several hours, due to the complexity of Markov analysis for a large number of system components even after the fault tree modularization.

Potentially the Markov model size for maintenance dependency type could be reduced by increasing the number of maintenance engineers. That would

result in the dependent component elimination from the model for each additional engineer. This factor could be easily implemented into the optimisation code. On the other hand, additional maintenance staff would cause the increase of the system life cycle cost. Hence, such an improvement measure should be considered carefully for each potential system design.

References

1 Cantoni M., Marseguerra M., and Zio E, Genetic Algorithms and Monte Carlo Simulation for Optimal Plant Design, *Reliability Engineering and System Safety* **68**, pp. 29-38 (2000).

2 Busacca P.G., Marseguerra M., and Zio E, Multiobjective Optimization by Genetic Algorithms: Application to safety systems, *Reliability Engineering and System Safety* **72**, pp. 59-74 (2001).

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

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

5 Everson R.M. and Fieldsend J. E, Multiobjective Optimization of Safety Related Systems: An Application to Short-Term conflict Alert, *IEEE Transactions on Evolutionary Computation*, University of Exeter, UK, pp. 1-12 (2006).

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

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

8 Andrews J. D., and Moss T. R., *Reliability and Risk Assessment*, Second Edition, Professional Engineering Publishing (2002).

9 A. Rauzy, New Algorithm for Fault Tree Analysis, Reliability Engineering and System Safety, **40**, pp. 203-211 (1993).

10 Sun H, System Dependency Modelling, PhD thesis, Dept. of Mathematical Sciences, Loughborough University, Loughborough, UK (2006).

11 Sbalzarini I. F., Muller S., and Koumoutsakos P., Multiobjective optimization using evolutionary algorithms, Center for Turbulence Research, *Proceedings of the Summer program 2000*, pp. 63-74 (2000).

12 Andrews J. D., and Pattison R. L., Genetic Algorithms in Optimal Safety System Design, *Proc. Instn. Mech. Engrs*., **213**, pp. 187-197 (1999).