Safety system design optimisation using a multi-objective genetic algorithm

Int. J. Reliability and Safety, Vol. X, No. Y, XXXX

Safety system design optimisation using a multi-objective genetic algorithm

Jelena Riauke and Lisa Bartlett*

Department of Aeronautical and Automotive Engineering,
Loughborough University,
Loughborough, LE11 3TU, UK
Email: J.Riauke@yahoo.co.uk Email: L.M.Bartlett@lboro.ac.uk
*Corresponding author

Abstract: This paper describes a design optimisation process applied to systems that require a high likelihood of functioning on demand. It is imperative that the best use of the available resources is made and an optimal rather than just an adequate system design is produced. The contribution of this research is in the development of an integrated approach which not only considers the primary system objective, availability of the system, but caters for all critical factors imperative to obtain an optimal system design. This research therefore combines the latest advantages of the fault tree analysis technique and the binary decision diagram method along with a multi-objective optimisation approach. The application area is a High Integrity Protection System of an offshore platform. The optimisation criteria involves unavailability, cost, spurious trip frequency and maintenance down time. The results produced using this method are compared to those obtained by exhaustive search.

Keywords: safety systems; unavailability; optimisation; genetic algorithms.

Reference to this paper should be made as follows: Riauke, J. and Bartlett, L. (xxxx) 'Safety system design optimisation using a multi-objective genetic algorithm', *Int. J. Reliability and Safety*, Vol. x, No. y, pp.xxx–xxx.

Biographical notes: Jelena Riauke is a final year research student working in the Systems, Risk and Reliability research group within the Aeronautical and Automotive Engineering Department at Loughborough University. She gained a BSc (2002) and MSc with Honours (2004) in Applied Mathematics from Kaunas University of Technology, Lithuania. Her PhD research is focused on safety system analysis and multi-objective optimisation by genetic algorithms.

Lisa Bartlett is a Lecturer in the Department of Aeronautical and Automotive Engineering at Loughborough University. She is one of five academic members of staff within the Reliability Research Group of the department. She gained her PhD in Fault Tree Analysis methods in 2000 from Loughborough University. Her PhD research focused on the Binary Decision Diagram approach, an alternative analysis method for fault tree analysis. Her current research interests are in safety system design optimisation, binary decision diagrams and fault diagnostics.

1 Introduction

Safety systems are designed to operate when certain conditions occur and to act to prevent their development into a hazardous situation. Failure of such a system or process may have severe consequences, possibly injuring members of the work force or public and occasionally resulting in loss of life. To minimise the likelihood of a hazardous situation, safety systems must be designed to minimise their unavailability.

The majority of safety systems involve objective functions and constraints that are too complicated to manipulate using linear programming and classical optimisation techniques. The modern heuristic optimisation techniques (Rao, 1996) have proved to be more efficient and preferable for safety systems optimisation, which have integer variable design parameters, small search space regions, and linear and nonlinear objective function characteristics. Nowadays one of the most powerful optimisation method groups is Genetic Algorithms (GAs) (Goldberg, 1989). Other efficient techniques are Great Deluge, Threshold Accepting and Particle Swarm Optimisation (Rao, 1996).

During the last decade a number of researchers have applied various methods for different safety system optimisations. Cantoni et al. (2000) used a simulation approach for optimal industrial plant design (to determine the choice of system layout and components) under conflicting safety and economic constraints. Marseguerra et al. (2004) proposed the multi-objective optimisation scheme for nuclear safety systems based on the effective coupling of genetic algorithms (MOGA) and Monte Carlo simulation. Martorell et al. (2004) considered a multiple-optimisation problem, where the parameters of design, testing and maintenance act as the design considerations. This problem was solved by several methods, with the best results obtained by the SPEA2-based MOGA. Everson and Fieldsend (2006) introduced the multi-objective optimisation based on GAs of safety related and critical systems. This research and others have shown the capability of the multi-objective approach and is the focus of this paper.

This paper describes a design optimisation tool which yields an optimal safety system design by fully utilising available resources. The novel attributes of the method include the integration of reliability and optimisation methods into one automated tool. The High Integrity Protection System (HIPS) of an offshore platform has been chosen to test the effectiveness of the proposed optimisation technique, which combines the fault tree (Andrews and Moss, 2002), binary decision diagram (Rauzy, 1993) and multi-objective evolutionary methods. The paper considers the HIPS optimisation by the multi-objective improved Strength Pareto Evolutionary Approach (SPEA2) (Zitzler, 2001) and compares results to those obtained by exhaustive search. Due to the multi-objective GAs universality, these methods often take less time to find the optimal solution than other multi-objective approaches and also require less computer resources (Sbalzarini et al., 2000). This advantage of the MOGAs helps to produce optimal results in a few minutes, which is very important for large safety systems. Considering these multi-objectives, with the results obtained for this application, show the advantages for safety system design. The algorithm developed has the potential to be applied to any safety system thus enabling effective analysis across a range of industrial domains.

The remainder of this paper is divided into four sections. The first overviews the optimisation technique. The second considers analysis of the example system and the implementation of the SPEA2 algorithm to the HIPS optimisation problem. The third discusses the results obtained from both the optimisation scheme and exhaustive search. The main conclusions are given in the final section.

2 Optimisation technique

The proposed optimisation technique can be described as a combination of fault tree analysis techniques, binary decision diagrams and the improved SPEA2. The fault trees are used to represent the system failure logic for each potential design (Section 2.1). Binary decision diagrams (Section 2.2) help to quantify the top event characteristics: unavailability and spurious trip frequency. The SPEA2 has been chosen for the system optimisation (Section 2.3).

2.1 Fault tree analysis

Fault Tree Analysis, first conceived in 1961 (Andrews and Moss, 2002), is one of the most powerful and widely used analytical techniques in the field of reliability engineering. It provides a well-accepted means of predicting the reliability of complex systems within the design optimisation algorithm. As no explicit objective function exists, fault trees are used to quantify the system unavailability. Each potential system design has an individual fault tree structure. However, it is an impractical task to construct separate fault trees for each design variation. This problem can be solved by including house events in the fault tree structure. House events are used to model two state events which either occur or do not occur, and, therefore, have probabilities 1 or 0. They provide a very effective means of turning sections of the fault tree on and off. One of the advantages of this is that the same fault tree can be used to model several scenarios.

Figure 1 Example fault tree with house events

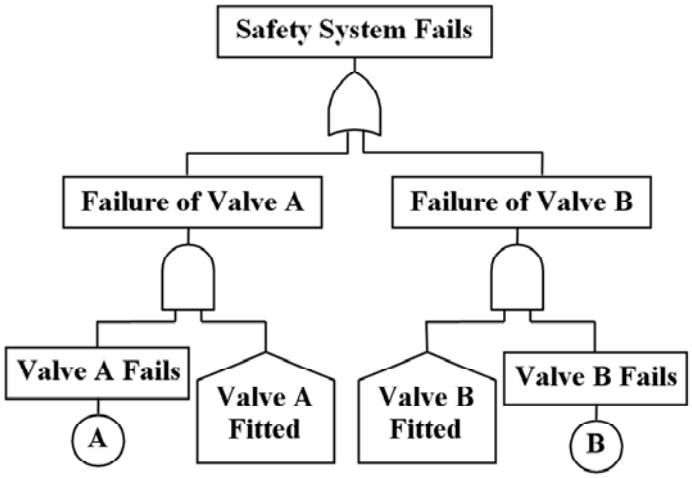


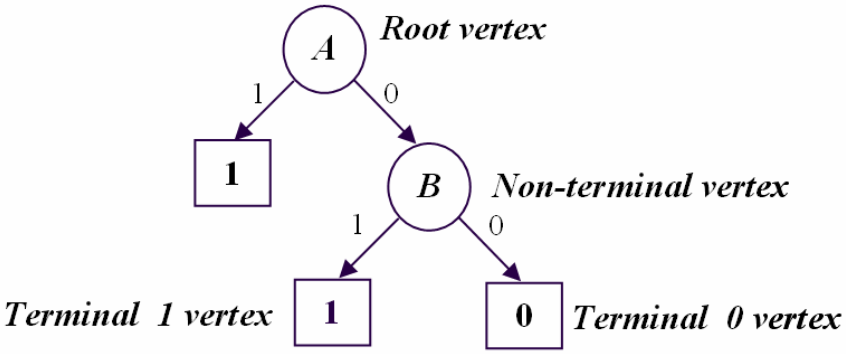
Figure 1 illustrates an example of a simple safety system, whose design may include two valves (*A* and *B*). The top event occurs if at least one of the valves fails. The house events ('Valve *A* fitted' and 'Valve *B* fitted') are used to represent the system design options. For example, if only valve *A* is fitted in the system, then the house event 'Valve *A* fitted' is set to true, i.e. assigned a probability of 1, 'Valve *B* fitted' is set to false, i.e. assigned a probability of 0. Therefore, the only way the safety system fails is from the contribution of valve *A* failing itself.

2.2 Binary decision diagrams

The conversion of the fault tree to the Binary Decision Diagram (BDD) format improves both the efficiency of determining the minimal cut sets of the fault tree and also the accuracy of the calculation procedure used to determine the top event parameters.

A BDD can be described as a rooted, directed acyclic graph (Rauzy, 1993). Consider the example fault tree from Figure 1. If both house events 'Valve *A* fitted' and 'Valve *B* fitted' are true, i.e. have probability 1, the resulting binary decision diagram is as shown in Figure 2. All paths through the BDD start at the root vertex (*A*) and terminate in one of the two states, either 1 or 0. State 1 corresponds to the system failure, state 0, conversely, corresponds to a system success. Each BDD is composed of vertices, connected by branches, which are divided into terminal and non-terminal. Non-terminal vertices correspond to the basic events of the fault tree, i.e. vertex *B* for the example BDD (Figure 2). Vertices 1 and 0 are terminal. Qualitative and quantitative analysis of the BDD is described in detail by Rauzy (1993).

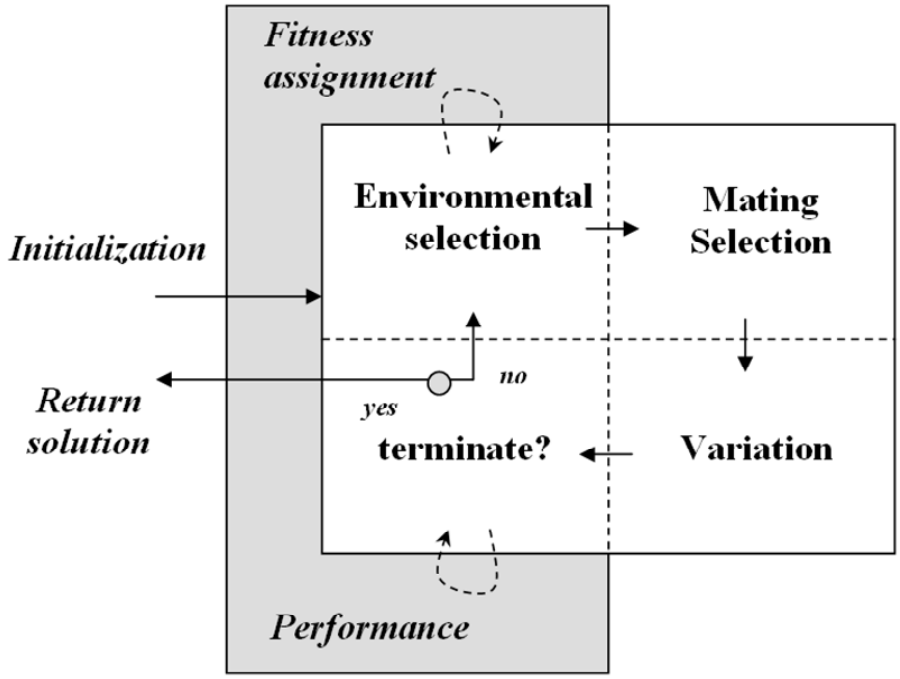
Figure 2 Example binary decision diagram



2.3 Multi-objective genetic algorithms

SPEA2 is a relatively recent evolutionary technique for finding or approximating the optimal solution set for multi-objective optimisation problems. It was designed by Zitzler et al. (2001) and is an improved version of the (SPEA), developed by Zitzler and Thiele in 1998. Figure 3 represents the schematic of the SPEA2.

Figure 3 The schematic of the SPEA2



The suggested algorithm (Figure 3) can be explained in six steps:

Step 1 – Initialisation: Generate an initial population of potential designs and create the empty archive called external set. The resultant archive after the optimisation is complete will hold the set of best designs.

Step 2 – Fitness assignment: Calculate the fitness value of each potential design in the initial population. This fitness value represents the suitability of the design given by the optimisation criteria.

Step 3 – Environmental selection: Copy all non-dominated designs to the archive (given the optimisation is a minimisation problem, the non-dominated solutions are those which have at least one smallest optimisation parameter value). If the archive is exceeded reduce it by means of the truncation operator, otherwise fill the archive with dominated designs from the initial population. The number of designs contained in the archive is to remain constant over time.

Step 4 – Termination: If the maximum number of generations is reached or another stopping criterion is satisfied then the set of possible designs are those in the archive. Algorithm complete. Else continue to Step 5.

Step 5 – Mating selection: Perform binary tournament selection with replacement on the archive in order to fill the mating pool (group of designs upon which genetic modification may occur), i.e.:

(a) Randomly (using uniformly distributed random numbers) select two individuals out of the archive.

(b) Copy the one with the better fitness value (i.e. lower for the HIPS optimisation problem) to the mating pool.

(c) If the size of the mating pool is equal to the size of the archive, then stop, else go to Step (a).

Step 6 – Variation: Apply recombination and mutation operators to the mating pool and set the archive to the resulting population (recombination is a process in which individual strings are copied according to their fitness values, and mutation is an operation that provides a random element in the search process). Increment generation counter and go to Step 2.

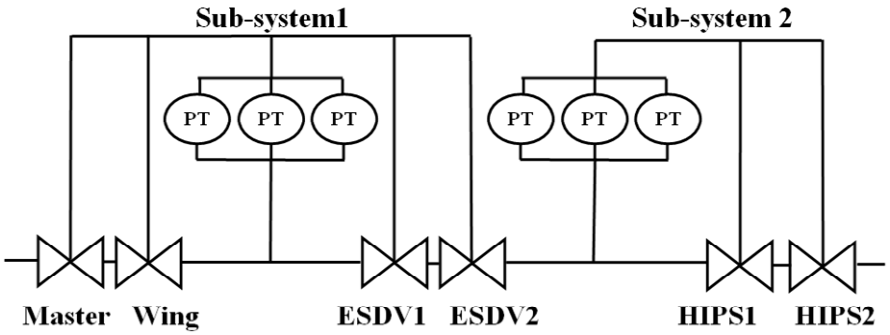
3 Example safety system

The High Integrity Protection System (HIPS) has been chosen to be optimised by the proposed technique. The structure of the system, failure data and design limitations are discussed in Section 3.1. Section 3.2 shows the application of the developed technique to the HIPS.

3.1 HIPS structure

The main function of the HIPS is to prevent a high-pressure surge passing through it. Protection is provided for processing equipment whose pressure rating could be exceeded. The high pressure originates from a production well of a not normally manned offshore platform and the pieces of equipment to be protected are located downstream on the processing platform. Figure 4 represents the main features of the HIPS (Andrews and Pattison, 1999).

Figure 4 Structure of high-integrity protection system



HIPS is divided into two separate subsystems. Subsystem 1 is the Emergency Shutdown or ESD subsystem. This is the first level of protection of the HIPS. The ESD system acts to close the Wing and Master valves together with any ESD valves that have been fitted when pressure in the pipeline exceeds the permitted value. This value is monitored using Pressure Transmitters (PT). Subsystem 2 provides an additional level of protection. Inclusion of the high-integrity protection system incorporates this second level of redundancy. The latter sub-system is completely independent in operation. Its method of protection is the same as the ESD system.

The HIPS is a relatively simple safety system yet the problem has ten main design variables. The number of potential designs considering these variables is 232, 257 and 600. It is a complex task to understand the interaction between all the design variables and is practically impossible for any design engineer to do by hand. Hence, an optimisation technique is required. The ten design variables, their description and evaluation limits are shown in Table 1.

Table 1 Main HIPS variables

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

It is assumed in the analysis that whatever valve type is selected all valves within the systems are fitted as this type. This is true of the pressure transmitter type also. In addition, the number of pressure transmitters required to activate the closure of valves on subsystem 1 or 2 is a function of the number installed (N_1 , N_2).

Failure data: Each component of the HIPS 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. Table 2 shows the failure rate and mean repair time for each HIPS component in both dormant and spurious failure modes. This data will be used subsequently when calculating the unavailability and spurious trip probability of the HIPS.

Design limitations: Each combination of HIPS variables gives a new system design. The choice of system design is not unlimited. In this case, there are three limitations 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. Hardware costs for each component in the system as well as times taken to service each component at each maintenance test are shown in Table 2.

Table 2 HIPS component failure data

Component	*Dormant Failure Rate (p/h)*	*Spurious Failure Rate (p/h)*	*Cost*	*Test Time (h)*
Wing Valve	1.14×10^{-5}	1×10^{-6}	100	12
Master Valve	1.14×10^{-5}	1×10^{-6}	100	12
HIPS Valve 1	5.44×10^{-6}	5×10^{-7}	250	15
HIPS Valve 2	1×10^{-5}	1×10^{-5}	200	10
ESDV Valve 1	5.44×10^{-6}	5×10^{-7}	250	15
ESDV Valve2	1×10^{-5}	1×10^{-5}	200	10
Solenoid Valve	5×10^{-6}	5×10^{-7}	20	5
Relay Contacts	0.23×10^{-6}	2×10^{-6}	1	2
PT 1	1.5×10^{-6}	1.5×10^{-5}	20	1
PT 2	7×10^{-6}	7×10^{-5}	10	2
Computer Logic	1×10^{-5}	1×10^{-5}	20	1
Mean Repair Time = 36 hours				

3.2 HIPS optimisation

The objective of the design optimisation problem for the HIPS application system is to minimise four system optimisation parameters (unavailability (Q_{sys}), spurious trip frequency (F_{sys}), cost (*Cost*) and maintenance down time (*MDT*)) by manipulating the design variables such that limitations placed on them by constraints are not violated. Constraints involved in this problem fall into the category of either explicit or implicit constraints. The cost and maintenance down time can be represented by an explicit function of the design parameters. On the other hand, the system unavailability and the number of spurious trips can only be calculated by a full analysis of the system. The fault tree analysis techniques combined with binary decision diagrams for quantification are implemented.

The C++ package was used to build the HIPS optimisation software called ISPEASSOP (Improved Strength Pareto Evolutionary Algorithm Safety System Optimisation Procedure). There are two main parts of the ISPEASSOP program. Part 1 is responsible for the HIPS structure and evaluation of the HIPS unavailability and spurious trip frequency. Part 2 is an implemented SPEA2 algorithm for the HIPS optimisation.

Part 1 – HIPS structure: 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 design options.

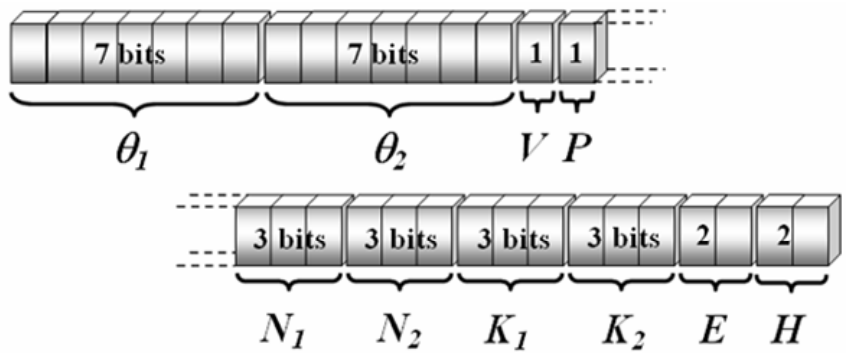
The spurious trip frequency for each design is also an implicit constraint that requires the use of fault tree analysis to assess its value. House events are again used to construct a fault tree capable of representing each potential design for this failure mode. The causal relationship 'HIPS fails spuriously' 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.

The corresponding house events within the fault tree are set to true or false for each design. The reduced fault tree is converted to the BDD for quantitative analysis. The probability values obtained from the analysis of the unavailability and spurious trip BDDs are used within the optimisation algorithm to select the best designs.

Part 2 – Optimisation algorithm: The initial population (Step 1 of the algorithm) consisted of 20 strings. Each string represents a particular system design depending on the values assigned to each of its 10 parameters (Table 1), where each parameter is calculated according to the binary coding system.

Each parameter 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. For example, the parameters governing the maintenance test interval for subsystems 1 and 2, θ_1 and θ_2 respectively, require 14 bits (7 bits each) of the total string to accommodate the maximum time span of 104 weeks each. 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 (Figure 5).

Figure 5 Binary representation of solution string



The restricted range of values assigned to each parameter does not in each case correspond to the representative binary range on the solution string. For this reason a specialised procedure is used to code, to initialise and to check the feasibility of strings at each optimisation step. In the initialisation step feasible strings are randomly regenerated.

Step 2 of the algorithm requires fitness assignment. Each fitness evaluation is dependent on the number of constraints: explicit and implicit. Explicit ones can be determined and easily evaluated from an explicit function of the design variables. Cost of the HIPS design is an explicit constraint and is represented by equations (1–3):

$$Cost = Cost(subsys1)+Cost(subsys2) \leq 1000, \tag{1}$$

$$\begin{aligned} Cost(subsys1) = {} & E(V_1C_{VE1} + V_2C_{VE2} + C_s) \\ & + N_1(P_1C_{P1} + P_2C_{P2}) + 261, \end{aligned} \tag{2}$$

$$\begin{aligned} Cost(subsys2) = {} & H(V_1C_{VH1} + V_2C_{VH2} + C_s) \\ & + N_2(P_1C_{P1} + P_2C_{P2}) + 21 \end{aligned} \tag{3}$$

where $C_{VE1} = C_{VH1}$ is the cost of the valve type 1, $C_{VE2} = C_{VH2}$ is the cost of the valve type 2, C_{P1} is the cost of the PT type 1, C_{P2} is the cost of the PT type 2, and C_s is the cost of the solenoid valves. The constant 261 [equation (2)] and 21 units [equation (3)] are 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 [equations (4), (5) and (6)]:

$$MDT = MDT(subsys1) + MDT(subsys2) \leq 130, \tag{4}$$

$$MDT(subsys1) = \frac{52}{\theta}[E(V_1 M_{VE1} + V_2 M_{VE2} + M_S) + N_1(P_1 M_{P1} + P_2 M_{P2}) + 47], \tag{5}$$

$$MDT(subsys2) = \frac{52}{\theta}[H(V_1 M_{VH1} + V_2 M_{VH2} + M_S) + N_2(P_1 M_{P1} + P_2 M_{P2}) + 13]. \tag{6}$$

where $M_{VE1} = M_{VH1}$ is the test time of the valve type 1, $M_{VE2} = M_{VH2}$ is the test time of the valve type 2, M_{P1} is the test time of the pressure transmitter 1, M_{P2} is the test time of the pressure transmitter 2 and M_S is the test time of the solenoid valve. The expression $52/\theta_i$ [equations (5) and (6)] gives the number of times the system is down in a year. The constant 47 [equation (5)] and 13 units [equation (6)] represents the sum of the test times for the fixed components in each subsystem.

The constraints are incorporated into the optimisation by penalising the unavailability when violation occurs, as the most important factor for system functionality is to work on demand [the constraint penalties are explained in detail by Andrews and Pattison (1999)]. Therefore, the overall unavailability of each string consists of four parts:

1 probability of the system failure, unavailability, Q_{sys}

2 penalty for exceeding the total cost constraint, C_p

3 penalty for exceeding the total maintenance down time constraint, M_p

4 penalty for exceeding the spurious trip constraint, S_p.

Each penalty is subsequently added to the system unavailability. The resulting value is a penalised system unavailability Q'_{sys}, which participates in the optimisation procedure:

$$Q'_{sys} = Q_{sys} + C_p + M_p + S_p. \tag{7}$$

Fitness assignment requires the division of the population of designs into dominated and non-dominated groups according to the following rules: since the optimisation is a minimisation problem, the design *a* dominates the design *b* if all *a* parameter values are equal to or smaller than *b* parameter values and at least one of parameter *a* values is smaller than the respective *b* parameter value.

The design a is non-dominated if there is no design in the population which dominates a. To avoid the situation that designs dominated by the same archive members have identical fitness values, for each individual both non-dominating and dominated solutions are taken into account. In detail, each design i in the archive and the population 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 the strengths of its dominators in both the archive and population. Although the raw fitness assignment provides a sort of niching mechanism based on the concept of Pareto dominance, it may fail when most designs do not dominate each other. Hence, 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 (Zitzler et al., 2001), 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 (8):

$$\sigma_{ij} = \sqrt{(C(i)-C(j))^2 + (MDT(i)-MDT(j))^2 + (Q^{'}(i)-Q^{'}(j))^2 + (Fsys(i)-Fsys(j))^2} \tag{8}$$

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 penalised 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, 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

$$D(i) = \frac{1}{\sigma_i^k + 2}. \tag{9}$$

In the denominator, 2 is added to ensure that its value is greater that zero. Finally, adding $D(i)$ to the raw fitness value $R(i)$ of the design i yields its fitness, *Fitness* (i):

$$Fitness(i) = R(i) + D(i)\,. \tag{10}$$

The last step of the SPEA2 requires the application of the crossover and mutation operators to the mating pool. There are many different types of the crossover operator. The standard SPEA2 algorithms uses single-point crossover (Goldberg, 1989) which works through the following scheme:

Step 1: A 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 $j + 2$.

Step 3: If population end is not reached the process repeats from step one for the next string in the population.

A ‘modified’ method has been created for this application. It is similar to single-point crossover. The main difference appears at the third step, when consideration is given to the second parent string from the pair. This string can again participate in crossover as the first parent. This modification developed through progressive research results in diversity of the produced strings and, therefore, makes the search for the optimal solution faster.

4 Results

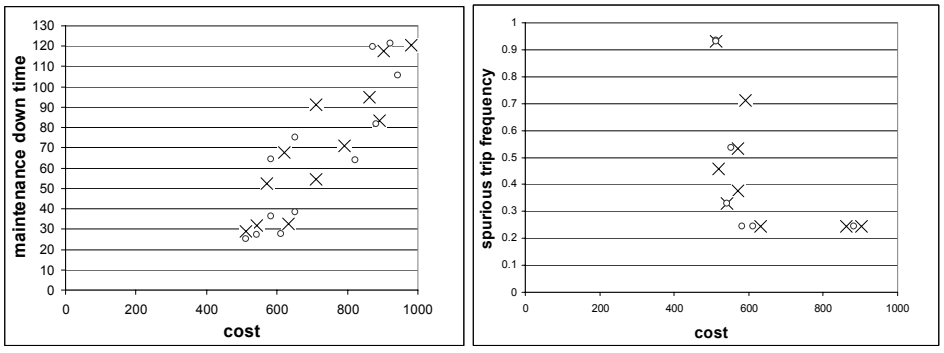
4.1 Optimisation results

Two different optimisation schemes have been implemented to tailor the algorithm parameters for the HIPS 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 30 different initial populations with only 100 generations for each run of the ISPEASSOP with the same crossover and mutation rates.

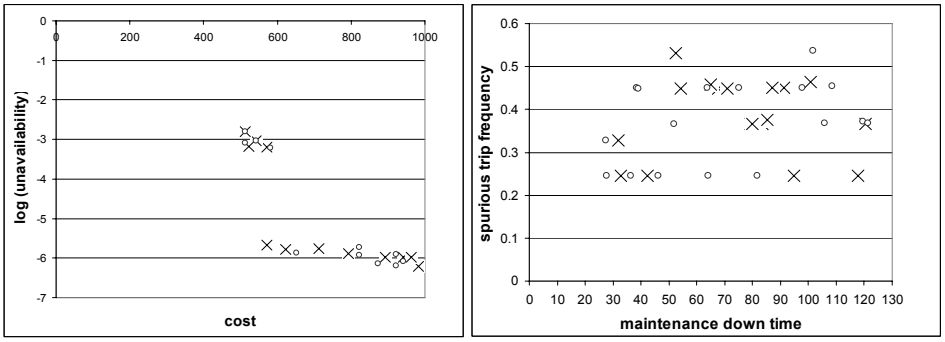
The first scheme resulted in a single Pareto set of non-dominated HIPS design options, on the other hand the second scheme gave 30 sets. A Pareto set obtained from the 20th run in the second scheme consisted of a larger number of non-dominated solutions by most optimisation parameter values and, therefore, has been chosen for comparison with the first optimisation scheme. Due to a large number of optimisation parameters the analysis of these results has been carried out for each group of two. Figures 6 (a–f) show the comparison of resulting Pareto fronts in two-dimensional space (where scheme 1 points are represented using a × and a° for scheme 2).

All the Figure 6 plots show that both optimisation 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 1st scheme produces up to 4 additional non-dominated designs. Figure 6(a) indicates that the higher maintenance down time values are directly proportional to the system cost values. Figure 6(b) illustrates that with an increase in cost comes a decrease in spurious trip frequency, however, at just below 600 cost units further increase does not improve the reduction in inappropriate action of the system. Figure 6(c) shows the relationship between cost and unavailability. A log scale has been used to represent the unavailability due to the large deviation in values produced. As cost is increased the effect is to improve the unavailability values. Higher maintenance down time values lead to a smaller system unavailability [Figure 6(e)]. Figure 6(d) shows that there is no clear relationship between spurious trip frequency maintenance down time. Figure 6(f) shows that there is again not a strong relationship between spurious trip and unavailability, however, values of 0.45 spurious trip occurrences per year produces the smallest unavailability.

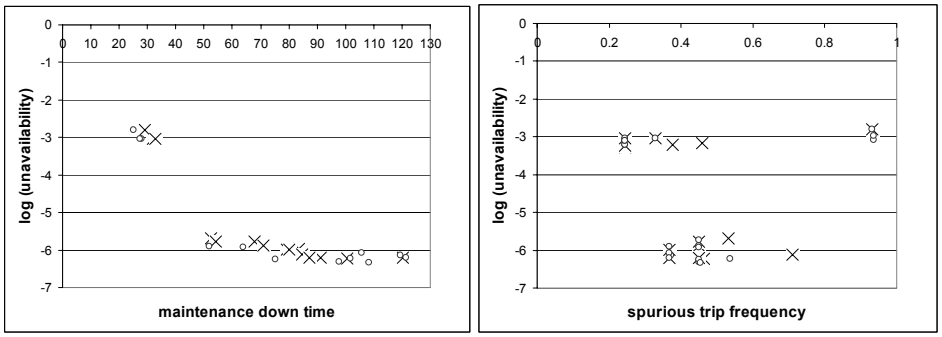
Figure 6 Pareto front in two-dimensional space



(a) Cost and MDT Pareto front (b) Cost and Fsys Pareto front



(c) Qsys and Cost Pareto front (d) Fsys and MDT Pareto front



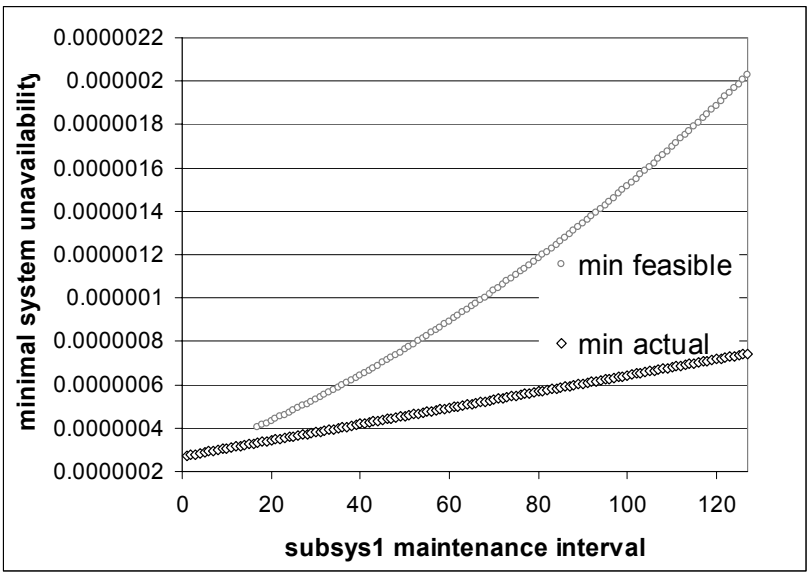
(e) Qsys and MDT Pareto front (f) Qsys and Fsys Pareto front

4.2 Exhaustive search results

For a system required to work on demand the system unavailability is one of the most important optimisation criterion. Therefore, in order to check the performance of the schemes from Section 4.1 in terms of system unavailability, an exhaustive search has been produced. Given the total number of potential HIPS designs equals 232, 257, 600, the whole exhaustive search required almost 38 hours. Evidence showed a difference

between the values of the unavailability obtainable with constraints removed. Figure 7 represents the comparison of the smallest actual (with constraints removed) and feasible (obtained from the design, with optimisation parameters within their limitations, discussed in Section 3.1) for system unavailability obtained during the search when considering maintenance test intervals for subsystem 1.

Figure 7 Comparison of the actual and feasible minimal Qsys obtained by exhaustive search



As it might be seen from Figure 7, the smallest actual system unavailability (*Qsys* = 2.884e-7) has been obtained in the first group ($\theta 1$ = 1 week), however, the design corresponded to this value is infeasible due to a large value of maintenance down time (*MDT* = 2028 *h*). The value of the smallest system unavailability has the tendency to increase when the maintenance interval $\theta 1$ becomes larger. The smallest feasible unavailability value is *Qsys* = 4.051e-7. The difference between the actual and feasible values increases when the maintenance interval becomes larger.

The comparison of the best results obtained during experiments described in Section 4.1 to the best HIPS design obtained by exhaustive search is shown in Table 3. This table shows that both optimisation schemes produced designs with unavailability values close to the one obtained by exhaustive search, however, the second scheme resulted in a smaller value. Other optimisation parameter values are similar in all cases. The first scheme produces smaller MDT (128.40 *h*) and system cost (632 units) values, on the other hand the second scheme resulted in smaller spurious trip frequency (0.45027 times). All three designs are very similar. In all cases the first subsystem has no ESD valves (*E*) and the subsystem 2 consists of only 1 HIPS valve (*H*). The dominated value for pressure transmitters fitted (*N1/N2*) and required (*K1/K2*) is 2. All designs are constructed of the first type of valve (*V*) and pressure transmitter (*P*) type. The maintenance interval for subsystem 1 ($\theta 1$) is close to 17, and the maintenance interval $\theta 2$ varies from 60 to 127. Other optimisation parameter values are also very similar for all three designs. Since the optimisation technique searches for the optimal solutions in

terms of four objectives, the likelihood of the resulting design with the smallest global unavailability is reduced. However, the obtained results prove the good performance of the developed tool, which produces an optimal solution close to the global minimum in only 20 minutes (for 3000 system evaluations).

Looking at the unavailability values for each of the optimisation schemes shows that in both experiments the majority of unavailability values are close to the feasible smallest value produced by the exhaustive search. 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. Despite the relatively small number of generations, the second optimisation scheme provided larger diversity between potential HIPS designs and leaded to a smaller system unavailability.

Table 3 Results comparison

	Subsystem	*Exhaustive Search*	*1st Scheme*	*2nd Scheme*
1	No. of ESD valves, E	0	0	0
	No. of PTs, N_1	2	2	3
	No. of PTs to trip, K_1	2	2	2
	Maintenance test interval, θ_1	17	19	18
2	No. of HIPS valves, H	1	1	1
	No. of PTs, N_2	2	2	3
	No. of PTs to trip, K_2	2	2	2
	Maintenance test interval, θ_2	127	60	93
	Valve type (V)	1	1	1
	PT type (P)	1	1	1
	MDT	129.53	128.40	129.53
	Cost	632	632	652
	Spurious trip occurrence (F_{sys})	0.45045	0.45044	0.45027
	System unavailability (Q_{sys})	4.051e-7	4.235e-7	4.143e-7

5 Conclusions

The research has demonstrated the applicability of a multi-objective approach to safety system design optimisation, in particular in the offshore domain. Comparative analysis to a single objective approach has shown the benefits of addressing all objectives and ultimately highlights the way forward for future design optimisation analysis.

An automated robust design optimisation process has been developed. Integration of more traditional reliability techniques, with the optimisation algorithm, has produced a new optimisation tool. The adequacy of the system performance in terms of unavailability calculation is assessed using the fault tree analysis technique. The causes of failure for each possible design alternative are represented by a single fault tree by using house events. The use of the BDD technique allows the solution of the fault tree in the most efficient manner.

Two different schemes of SPEA2 have been successfully applied to a High Integrity Protection System (HIPS) and produced good results for system design optimisation. In both cases, the multi-objective approach produced a set of potential designs to be selected on the most important parameters. In both cases results were close to those obtained by exhaustive search. The first scheme provided a more accurate Pareto front due to a larger number of generations. On the other hand, the second scheme benefits from a larger diversity among possible design options and, therefore, leads to smaller optimisation parameter values.

Another important advantage of the SPEA2 is that it is fast and allows optimal solutions to be produced in only a few minutes time, hence, requires less memory resources.

References

Andrews, J.D. and Pattison, R.L. (1999) 'Genetic algorithms in optimal safety system design', *Proceedings of International Mechanical Engineers*, Vol. 213, pp.187–197.

Andrews, J.D. and Moss, T.R. (2002) *Reliability and Risk Assessment*, 2nd ed., Professional Engineering Publishing.

Cantoni, M., Marseguerra, M. and Zio, E. (2000) 'Genetic algorithms and Monte Carlo simulation for optimal plant design', *Reliability Engineering and System Safety*, Vol. 68, pp.29–38.

Everson, R.M. and Fieldsend, J.E. (2006) 'Multi-objective optimization of safety related systems: an application to short-term conflict alert', *IEEE Transactions on Evolutionary Computation*, University of Exeter, UK, pp.1–12.

Goldberg, D.E. (1989) *Genetic Algorithms in Search. Optimization and Machine Learning*, Addison-Wesley Publishing Company.

Marseguerra, M., Zio, E. and Podofillini, L. (2004) 'A multi-objective genetic algorithm approach to the optimization of the technical specifications of a nuclear safety system', *Reliability Engineering and System Safety*, Vol. 84, pp.87–99.

Martorell, S., Sanchez, A., Carlos, S. and Serradell, V. (2004) 'Alternatives and challenges in optimizing industrial safety using genetic algorithms', *Reliability Engineering and System Safety*, Vol. 86, pp.25–38.

Rao, S.S. (1996) *Engineering Optimization. Theory and Practice*, 2nd ed., John Wiley & Sons.

Rauzy, A. (1993) 'New algorithm for fault tree analysis', *Reliability Engineering and System Safety*, Vol. 40, pp.203–211.

Sbalzarini, I.F., Muller, S. and Koumoutsakos, P. (2000) 'Multi-objective optimization using evolutionary algorithms', Centre for Turbulence Research, *Proceedings of the Summer Program 2000*, pp.63–74.

Zitzler, E., Laumanns, M. and Thiele, L. (2001) 'SPEA2: improving the strength Pareto evolutionary algorithm', *Computer Engineering and Communication Network Lab (TIK)*, Swiss Federal Institute of Technology, TIK-Report No. 103.

Zitzler, E. and Thiele, L. (1998) 'An evolutionary algorithm for multi-objective optimization: the strength Pareto approach', *Computer Engineering and Communication Network Lab (TIK)*, TIK-Report No. 43.