This item was submitted to Loughborough's Institutional Repository by the author and is made available under the following Creative Commons Licence conditions.

creative commons

COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/

Safety System Optimization By Improved Strength Pareto Evolutionary Approach (SPEA2)

Jelena Borisevic and Lisa M Bartlett
Aeronautical and Automotive Engineering Department, Loughborough University, Loughborough, UK

Abstract

Safety systems are designed to prevent the occurrence of certain conditions and their future development into a hazardous situation. The consequence of the failure of a safety system of a potentially hazardous industrial system or process varies from minor inconvenience and cost, to personal injury, significant economic loss and death. To minimise the likelihood of a hazardous situation, safety systems must be designed to maximise their availability. The purpose of this paper is to describe a design optimization scheme using a multi-objective genetic algorithm applied to an offshore platform process. The optimization criteria involves unavailability, cost, spurious trip and maintenance down time to obtain an optimal safety system design.

Analyses of individual system designs are carried out using the latest advantages of the fault tree analysis technique and the binary decision diagram approach. The improved strength Pareto evolutionary approach (SPEA2) is chosen to perform the system optimization resulting in the final design specifications. The results produced using this method are compared to those using a single objective optimization approach. The overall conclusions show the benefit of using this technique for the application system.

1. Introduction

Safety systems installed on potentially hazardous plant require the maximum likelihood of working on demand. Therefore, it is imperative that the best use of available resources is made and an optimal not just adequate system design is produced.

The traditional engineering design process involves a trial and error type approach, where a design is created, analysed, and compared with a predetermined criterion of acceptability. These approaches produce a resulting system design that is usually adequate rather than optimal. To find an optimal system design a process is required which considers a number of design alternatives. It is highly unlikely that that the design parameters can be manually selected such that the optimal system performance is achieved within the available resources.

The majority of safety systems involve objective functions and constraints that are too complicated to manipulate using linear programming and classical optimization techniques. The modern heuristic optimization techniques [1], have proved to be the more efficient and preferable for safety systems optimization, which have integer variable design parameters, small search space regions, and linear and nonlinear objective function characteristics. Nowadays the most powerful optimization method group is genetic algorithms (GAs) [2]. Other efficient techniques are Great Deluge, Threshold Accepting and Particle Swarm Optimization [1].

During the last decade a number of engineers have applied various methods for different safety system optimizations. Cantoni [3] 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 [4] proposed the multiobjective optimization scheme for nuclear safety systems based on the effective coupling of genetic algorithms (MOGA) and Monte Carlo simulation. Martorell [5] considered a multiple-optimization problem, where the parameters of design, testing and maintenance act as the decision variables. This problem was solved by several methods, with the best results obtained by the SPEA2-based MOGA. Everson and Fieldsend [6] introduced the multi-objective optimization based on the 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.

Previous work has been performed on the high integrity protection system (HIPS) of an offshore platform, using simple GAs [7]. This paper considers improving the HIPS optimization procedure by adopting the improved strength Pareto evolutionary approach (SPEA2) [8], which is a multi-objective optimization technique. The technique is combined with the fault tree [9] and binary decision diagram [10] methods.

The remainder of this paper is divided into six sections. The first describes the high integrity protection system and the design considerations. The second considers analysis of the system. The third overviews the optimization technique. The fourth represents the implementation of the SPEA2 algorithm to the HIPS optimization problem, and final sections discuss the obtained results and conclusions.

2 The HIPS System

The system design to be optimised is the High Integrity Protection System (HIPS) [7]. 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 1 represents the main features of the HIPS.

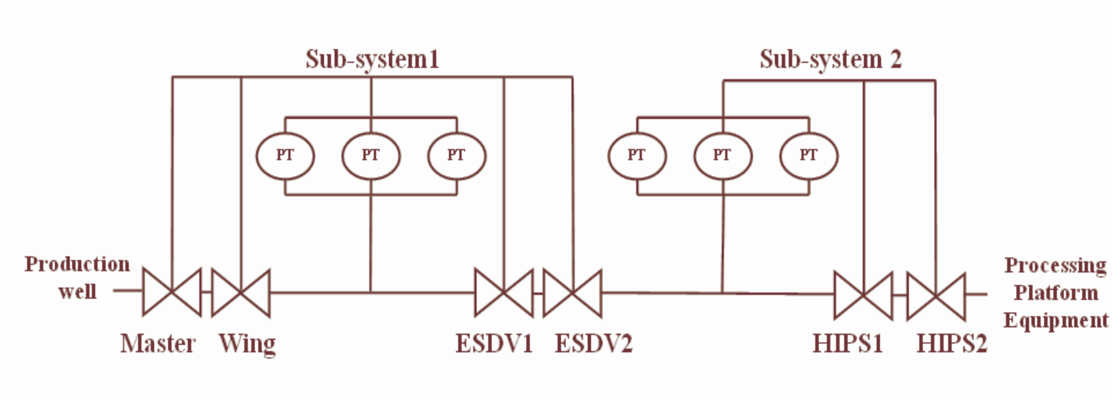

Figure 1. High Integrity protection system

HIPS is divided into two separate subsystems. Sub-system 1 is the Emergency Shutdown or ESD sub-system. 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).

Sub-system 2 provides an additional level of protection. Inclusion of the high-integrity protection system incorporates this second level of redundancy. An important fact is that the latter sub-system is completely independent in operation. Its method of protection is the same as the ESD system.

2.1 Main Design Variables

The HIPS is a relatively simple system, yet there are a huge number of design options which can be considered. Ten main design variables describe this particular system. These variables, their description and evaluation limits are shown in table 1.

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, 1, 2
H	Number of HIPS valves fitted	0, 1, 2

Table 1. Main HIPS Variables

It is assumed in the analysis that whatever valve type is selected all valves within the system 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).

The number of potential design variations considering just ten design variables is 45,158,400. It would be impractical to evaluate exhaustively each potential design. Furthermore, 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. A technique is required to determine the 'best' design option in a more practical manner. This is to be achieved using a computerised multi-objective optimisation algorithm.

2.2 Failure Data and Design Limitations

Each hardware 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. For the optimisation procedure information is provided on 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.

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 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. Hardware costs for each component in the system as well as times taken to service each component at each maintenance test are provided for the analysis.

3. System Analysis

The objective of the design optimization problem for the HIPS application system is to minimize four system optimization parameters (unavailability (Q_{sys}), spurious trip frequency (F_{sys}), cost and maintenance down time) 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 technique combined with binary decision diagrams for quantification are implemented.

3.1 Fault Tree Analysis

As no explicit objective function exists, fault trees are used to quantify the system unavailability of each potential design. However, it is an impractical task to construct a fault tree 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 [9]. 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 2 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 the valves fail. Two house events ("Valve *A* fitted" and "Valve *B* fitted") are used to represent the system design options.

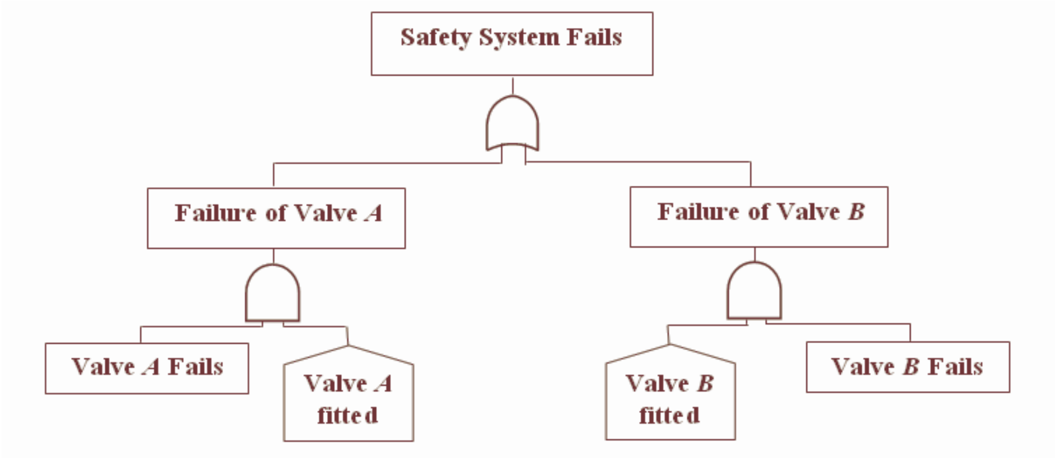

Figure 2. Example of the Fault Tree with House Events

As the system investigated is a safety system, its ability to function on demand or its availability is paramount. Therefore, the unavailability is one performance statistic used for determining the optimal design. 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.

3.2 Binary Decision Diagrams

The conversion of the fault tree to the 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 (Figure 3). 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. vertices *B*, *C* and *D* for the example BDD. Vertices *1* and *0* are terminal.

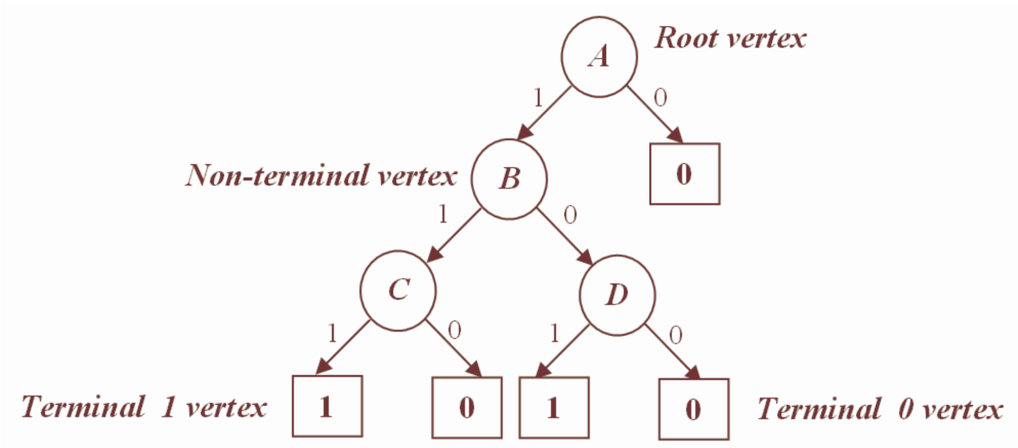

Figure 3. Example of the Binary Decision Diagram

The initial HIPS designs are created using the optimization approach. 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 optimization algorithm to select the best designs.

4. SPEA2 Overview

SPEA2, designed by Zitzler, Laumanns and Thiele [8], is an improved version of the strength Pareto evolutionary algorithm (SPEA), developed by Zitzler and Thiele in 1998 [11]. It is a relatively recent evolutionary technique for finding or approximating the optimal solution set for multiobjective optimization problems. SPEA2 has shown very good performance in comparison to other multiobjective genetic algorithms [8] and, therefore, has been selected for the HIPS optimization. The suggested algorithm can be explained in six steps:

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

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

Step 3. *Environmental selection*: Copy all nondominated designs to the archive (given the optimization is a minimization problem, the nondominated solutions are those, which have at least one smallest optimization 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 (i.e. lower for the HIPS optimization problem) fitness value 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*.

5. SPEA2 Implementation

The C++ package was used to build the HIPS optimisation software called ISPEASSOP (Improved Strength Pareto Evolutionary Algorithm Safety System Optimization Procedure). There are three main parts of the ISPEASSOP program. Part one is responsible for the HIPS structure, part two is responsible for analysis using the Binary Decision Diagram method which calculates the HIPS unavailability and spurious trip frequency, and part three is an implemented SPEA2 algorithm for the HIPS optimisation.

5.1 *Coding and Initializing the Population*

The number of strings for the initial population (step 1 of the algorithm) for a problem is not defined, thus, based on the HIPS optimization by simple GAs [9], initial research has used 20. 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 4).

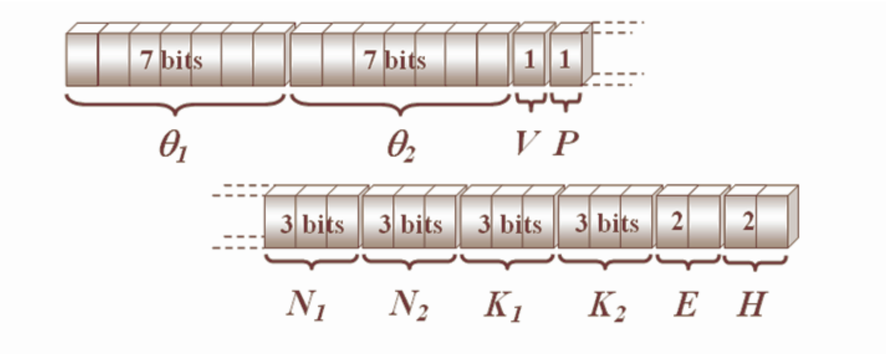

Figure 4. Binary Representation of Solution String

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 specialized procedure is used to code, to initialize and to check the feasibility of each string. In the initialization step infeasible strings are randomly regenerated.

5.2 *Optimization Parameters Evaluation*

Step two 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. In contrast, implicit constraints can only be evaluated by a full analysis of the system. Cost of the HIPS design is an explicit constraint and is represented by equations (1 - 3):

$$Cost = Cost(subsystem1) + Cost(subsystem2) \leq 1000, \tag{1}$$

$$Cost(subsystem1) = E(V_1C_{VE1} + V_2C_{VE2} + C_s) + N_1(P_1C_{P1} + P_2C_{P2}) + 261, \tag{2}$$

$$Cost(subsystem2) = H(V_1C_{VH1} + V_2C_{VH2} + C_s) + N_2(P_1C_{P1} + P_2C_{P2}) + 21, \tag{3}$$

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 subsystem 1 and subsystem 2 for each potential design (equations 4 - 6):

$$MDT = MDT(Subsystem1) + MDT(Subsystem2) \leq 130, \tag{4}$$

$$MDT(Subsystem1) = \frac{52}{\theta_1}\left[E\left(V_1M_{VE1} + V_2M_{VE2} + M_S\right) + N_1\left(P_1M_{P1} + P_2M_{P2}\right) + 47\right], \tag{5}$$

$$MDT(Subsystem2) = \frac{52}{\theta_2}\left[H\left(V_1M_{VH1} + V_2M_{VH2} + M_S\right) + N_2\left(P_1M_{P1} + P_2M_{P2}\right) + 13\right]. \tag{6}$$

where $M_{VE1} = M_{VH1}$ is a 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 $52/\theta$ (equations 5 - 6) gives the number of times the system is down in a year. The constant 47 (equation 5) and 13 units (equation 6) represent the sum of the test times for the fixed components in each subsystem.

The system unavailability and spurious trip frequency are calculated by setting to TRUE or FALSE corresponding house events in the fault tree given by the design parameters, then the BDD is formed and the required probability and frequency are calculated. Constraints are incorporated into the optimization by penalizing the unavailability when they are violated by the design (the constraint penalties are explained in detail in reference 9). 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 optimization 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 nondominated groups according to the following rules: since the optimization is a minimization 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* value is smaller that the respective *b* parameter value.

The design *a* is nondominated 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 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 [10], 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 $C(i)$ is the cost of the i-th design, $Q(i)$ is the i-th designs penalized system unavailability, j is from the interval [1,.., 20] with the condition that $i \neq j$. Obtained distances are stored in a list or 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, two 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 $F(i)$:

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

6. Results

Several ISPEASSOP runs have been implemented to tailor the algorithm parameters for the HIPS system (used for comparison purposes with the simple GAs results [7]). Tables 2 and 3 represent the characteristics of the fittest designs obtained after 10 runs of the ISPEASSOP (100 generations each). The chosen designs are nondominated by most optimization parameter values.

Run No.	**Cost**	**MDT**	F_{sys}	Q_{sys}
1	592	129.7008	0.455	4.5e-7
2	512	129.6974	0.332	8.33e-4
3	582	128.7361	0.324	6.8e-4
4	922	128.2273	0.718	1e-6
5	882	129.1590	0.166	1e-6
6	992	129.2523	0.552	1e-6
7	852	128.3286	0.245	6.55e-4
8	542	128.9881	0.324	8.45e-4
9	872	129.9032	0.377	1e-6
10	862	129.7309	0.999	1e-6
Average values	**761**	**129.1724**	**0.449**	**3.01e-4**

Table 2. Fittest Designs by ISPEASSOP after 10 Runs of 100 Generations

Design Variables										
Run No.	θ_1	θ_2	V	P	N_1	N_2	K_1	K_2	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 variable values for Table 2

Table 4 shows the fittest design produced by ISPEASSOP program after 10 runs and the best design obtained by single GAs [7] after 10 runs using the same parameter values (100 generations, 0.01 mutation rate and 0.7 crossover rate).

		GAs	**ISPEASSOP**
Subsystem 1	No. of ESD valves (E)	0	0
	No. of PTs (N_1)	2	1
	No. of PTs to trip system (K_1)	1	1
	Maintenance test interval (θ_1)	29	25
Subsystem 2	No. of HIPS valves (H)	2	1
	No. of PTs (N_2)	3	3
	No. of PTs to trip system (K_2)	2	3
	Maintenance test interval (θ_2)	32	73
Valve type (V)		2	1
PT type (P)		1	2
MDT		**128.43**	**129.7008**
Cost		**822**	**592**
Spurious trip occurrence (F_{sys})		**0.717**	**0.455**
System unavailability (Q_{sys})		**7.6e-4**	**4.5e-7**

Table 4. Results Comparison

Table 4 shows that the SPEA2 optimization algorithm, implemented in the ISPEASSOP program, gives better results in that the Q_{sys} is lower and all other parameters are within constraint limits. The available MDT resources are fully used (MDT is very closed to 130), the total system cost is smaller (a price reduction of 230 units) as well as the spurious trip occurrence, F_{sys} (approximately 1.5 times smaller).

In both optimization programs the maximum number of generations is equal to 100. The fittest design produced by the simple GAs is achieved only in the 70th generation. In contrast, in all 10 runs of 100 generations of the ISPEASSOP the fittest strings occurred in the first 10 generations. Consequently, the ISPEASSOP program requires less computer memory recourses, which is an important advantage for large safety systems optimization problems.

7. Conclusions

An automated robust design optimization process has been developed for the application to safety systems. The adequacy of the system performance in terms of availability calculation is assessed using fault tree analysis techniques. The causes of failure for each possible design alternative of a safety system is represented by a single fault tree by using the house events. The use of the BDD technique allows the solution of the fault tree in the most efficient manner.

The SPEA2 has been successfully applied to a high integrity protection system (HIPS) and produced good results for system design optimization. The SPEA2 produced improved results compared to those obtained by simple GAs. Another important advantage of the SPEA2 is that it is faster and requires less memory resources.

The HIPS is a relatively simple example of a safety system. Many systems are much more complex and have a much larger number of design variables. Therefore, the future work will be concentrated on testing the effectiveness of the technique on larger and more detailed safety system.

References

1. Rao S. S., *Engineering Optimization. Theory and Practice. Third Edition*, John Wiley & Sons (1996).
2. Goldberg D. E., *Genetic Algorithms in Search. Optimization and Machine Learning*, Addison-Wesley Publishing Company (1989).
3. Cantoni M., Marseguerra M., Zio E., *Genetic Algorithms and Monte Carlo Simulation for Optimal Plant Design*, Reliability Engineering and System Safety 68, pp. 29-38 (2000).
4. Marseguerra M, Zio E., 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).
5. Martorell S., Sanchez A., Carlos S., Serradell V., *Alternatives and Challenges in Optimizing Industrial Safety Using Genetic Algorithms*, Reliability Engineering and System Safety 86, pp. 25-38 (2004).
6. Everson R. M., 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).
7. Andrews J. D, Pattison R. L, *Genetic Algorithms in Optimal Safety System Design*, Proc. Instn. Mech. Engrs., Vol. 213, pp. 187-197 (1999).
8. Zitzler E., Laumanns M., 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).
9. Andrews J. D, Moss T. R., *Reliability and Risk Assessment*, Second Edition, Professional Engineering Publishing (2002).
10. Rauzy A., *New Algorithm for Fault Tree Analysis*, Reliability Engineering and System Safety, Vol. 40, pp. 203-211 (1993).
11. Zitzler E., Thiele L., *An Evolutionary Algorithm for multiobjective Optimization: The Strength Pareto Approach*, Computer Engineering and Communication Network Lab (TIK), TIK-Report No. 43 (1998).