IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

International Journal of Safety and Security Engineering
Vol. 13, No. 6, December, 2023, pp. 975-986
Journal homepage: http://iieta.org/journals/ijsse

Cost Effective Analysis of the Design of Safety Instrumented Systems Using Manta-Ray Foraging Optimization Algorithm

Fateh Berrah[1*], Mourad Chebila[2], Fares Innal[3], Adel Got[4]

[1] LRPCSI (Laboratory of Physico-Chemistry Research on Surfaces and Interfaces), Department of process Engineering, Faculty of Technology, University of 20Août 1955, Skikda 21000, Algeria
[2] LRPI (Laboratoire de Recherche en Prévention Industrielle), Institute of Health and Safety, University of Batna 2, Batna 05078, Algeria
[3] LAS (Laboratoire d'Automatique de Skikda), Institute of Applied Sciences and Techniques, University of 20Août 1955, Skikda 21000, Algeria
[4] LRIA (Laboratory for Research in Artificial Intelligence), Department of informatics, University of Science and Technology Houari Boumediene, Algiers, Algeria

Corresponding Author Email: f.berrah@univ-skikda.dz

https://doi.org/10.18280/ijsse.130601

Received: 20 June 2023
Revised: 16 November 2023
Accepted: 29 November 2023
Available online: 25 December 2023

Keywords:
safety instrumented system, cost effectiveness analysis, lifecycle cost, genetic algorithms, Manta ray foraging optimization

ABSTRACT

This study aims to develop a new cost effectiveness analysis framework in the context of safety instrumented systems (SIS) design and operation. The primary objective is to achieve an optimal equilibrium among safety integrity, operational integrity, and lifecycle cost of SIS. It is essential to note that these objectives may often be in conflict; for instance, enhancing safety integrity could potentially diminish operational integrity and escalate costs. Achieving this balance is crucial to ensure that the risk level being addressed aligns precisely with the desired objectives while minimizing any adverse effects. The novelty of this paper lies in the refined formulation of a multi-objective optimization problem and the application of a recently developed swarm-based Manta-Ray Foraging Optimization (MRFO) algorithm. The effectiveness of this approach is demonstrated through a typical SIS design challenge, which entails satisfying specific measures in terms of Safety Integrity Level (SIL), spurious trip activation rate, and lifecycle cost. These measures depend on variables such as the number and voting scheme of components, their types, and the intervals for potential proof tests. For validation and comparison, the problem was initially tackled using a conventional approach based on genetic algorithms. Subsequently, the MRFO algorithm was employed, yielding highly satisfactory results and confirming its proficiency in resolving real-world SIS optimization challenges. Notably, the MRFO algorithm produced a greater number of solutions compared to the genetic algorithm approach. This increase in solution options is advantageous, offering decision-makers a broader array of choices for optimal system design. This study contributes significantly to the field of SIS design, presenting an innovative, algorithm-driven approach to balancing safety, operational integrity, and cost in system development. It also contributes to understanding the life cycle costs of security barriers in general.

1. INTRODUCTION

The advent of significant technological and industrial advancements has been accompanied by the occurrence of major accidents, such as those in Seveso, Bhopal, and Piper Alpha. These incidents have underscored the necessity for comprehensive frameworks dedicated to the effective management of associated risks, encompassing a diverse array of processes, tools, and methodologies. Central to the protection of hazardous installations is the implementation of safety barriers, among which SIS is pivotal. SIS plays an instrumental role in detecting abnormal conditions, such as high pressure or gas leakages, and autonomously transitioning the equipment or installation to a safe state, such as process shutdown, thereby mitigating the escalation of process deviations into severe consequences with minimal or no human intervention. The criticality of SIS in ensuring safety is highlighted by incidents like the Buncefield disaster, predominantly attributed to the failure of an automatic overfilling system. This underscores the imperative for a robust framework to guide the effective design and operation of SIS, commensurate with the level of risk they are intended to mitigate. This necessity initially led to the development of the IEC 61508 functional safety generic standard [1], subsequently paving the way for sector-specific standards such as IEC 61511 for the process industry [2]. These standards delineate the requirements essential for ensuring the proficiency of SIS in executing their designated safety functions.

Designing a SIS that efficiently performs its risk-reducing functions necessitates the consideration of numerous factors. These factors include the behavior of the SIS under various conditions, as well as the requirements of the system being monitored and its environment. Beyond ensuring safety, the design process must also address potential operational disruptions that could arise from an unexpected activation of the SIS. Therefore, it is essential to strike a balance between the SIS's ability to ensure the safety of the protected equipment (referred to as safety integrity) and its capacity to operate without impeding normal functioning (operational integrity). This balance must be achieved at the lowest possible cost. Attaining this equilibrium is feasible through the application of cost-effectiveness analysis. This approach incorporates a lifecycle cost (LCC) model, which delineates the significant costs associated with the system's lifecycle, from design to decommissioning. The LCC model serves as a crucial tool in understanding and minimizing the expenses entailed in maintaining the SIS's functionality and integrity throughout its operational life.

The design and optimization of Safety Instrumented Systems (SIS) have garnered significant interest within the field. For instance, Torres-Echeverria [3] introduced two novel techniques for optimizing SIS, with a particular focus on testing policies. Additionally, Torres-Echeverria et al. [4] delved into the multi-objective optimization of proof testing policies using a genetic algorithm (GA). This approach quantitatively integrates the average probability of failure on demand (PFD_{avg}), spurious trip rate (STR), and LCC.

Furthermore, Torres-Echeverria et al. [5] explored the impact of component redundancy and diversification in SIS subsystem architectures, demonstrating enhancements in SIS performance during the design phase. Torres-Echeverria et al. [6] investigated the multi-objective optimization of SIS design and testing policies, using K out-off N (KooN) redundancy and the multi-objective genetic algorithm NSGA-II. The study undertook two distinct optimization cases: one focused on system design, encompassing component selection and redundancy allocation, and the other on testing policy optimization. In the study of Innal et al. [7], the challenge of optimizing SIS architecture design was initially approached through a preliminary search for a balance between performance measures, based on the analysis of KooN architectures. This was followed by a comprehensive approach utilizing GA to optimize various performance indicators along with maintenance and purchase costs. Lastly, Touahar et al. [8] targeted maintenance strategies aimed at optimizing SIS performance and minimizing spurious shutdowns during the operational phase. This methodology was applied to the emergency shutdown system of a blower section, showcasing the practical applicability of these GA-based approaches in real-world scenarios.

It is observed that a majority of studies in the field of SIS optimization predominantly employ GA. However, the literature reveals the existence of numerous alternative methods that exhibit competitively high performance. Additionally, a notable limitation in many of these studies is the omission of SIS design constraints, which can result in suboptimal or inefficient outcomes. This oversight highlights the need for a more comprehensive approach in SIS design optimization, one that not only leverages diverse algorithmic strategies but also thoroughly incorporates all relevant design constraints to ensure the efficacy and reliability of the optimized systems.

In this research, a more refined mathematical formulation of the SIS design problem is proposed, particularly with respect to the LCC. This includes the consideration of various cost factors beyond just maintenance and purchase. A significant contribution of this study is the application of the MRFO algorithm, a recently developed method, to address the SIS design optimization challenge. The MRFO algorithm has demonstrated commendable proficiency in handling single-objective real-world problems and has been adapted to multi-objective problems with linear and nonlinear constraints, as developed by Got et al. [9]. It is noteworthy that the application of the MRFO algorithm in the context of SIS or safety-related studies is unprecedented. To substantiate the efficacy of the results obtained through MRFO, comparisons are made with results derived using GA, ensuring that all relevant constraints are meticulously accounted for in the process.

The remainder of this paper is structured as follows. Section 2 is dedicated to the general presentation of SIS design problem, which involves many functional safety, LCC and cost-effectiveness analysis related concepts. Section 3 provides a presentation of MRFO and its application in the context of SIS. Section 4 gives an illustrative example using both MRFO and GA. Section 5 summarizes a few conclusions.

2. PROBLEM DESCRIPTION AND FORMULATION

A SIS, through its safety functions, should achieve the required risk reduction established during the risk analysis process (safety integrity) without disrupting the normal operation of the protected system in the absence of a dangerous situation (operational integrity). Obviously, if spurious emergency shutdowns are too frequent, they prove to be economically detrimental. Furthermore, these two quantities (safety and operational integrities) are antagonistic. Thus, attempting to increase safety integrity, by reducing dangerous failures of the SIS, can also significantly reduce its operational integrity by increasing nuisance trips (the converse is true). In addition, the different costs related to the SIS life cycle should be taken into account when trying to satisfy the two above-mentioned performances. Therefore, the best policy to design an effective SIS is that of an optimal compromise between its safety integrity, operational integrity and the potential costs throughout its life cycle. The following subsections detail the different contributing parameters to the SIS design problem.

2.1 Generalities about safety instrumented systems

Safety instrumented systems (SIS) are the basis of functional safety, whose importance and criticality necessitated the creation of common practices covering all the stages of their life cycle from the initial design until their decommissioning. Several international standards have been developed for this purpose including mainly the IEC 61508 generic standard [1], which covers the functional safety that can be ensured using Electrical / Electronic / Programmable Electronic (E/E/EP) systems, and the IEC 61511 standard [2] derived from the first one for the process industry sector.

The IEC 61508 defines a SIS as "*an E/E/PE system for safety applications that includes all system elements necessary to perform the safety function*", while the IEC 61511 considers that *"instrumented system used to implement one or more safety instrumented functions (SIFs) and a SIS consists of any combination of sensor(s), logic solver(s) and final element(s)*".

Therefore, a SIS aims to implement one or more functions to ensure or achieve a safe state of the equipment under control (EUC) in relation to a specific dangerous event. These functions are called "safety instrumented functions (SIFs)". A simple example of a SIF can be the opening of a coolant valve initiated by the rise in temperature of a reactor. These functions are measured using the concept of safety integrity level (SIL), where the SIL of a SIF represents the level of risk reduction it can or should achieve. In addition, the ability of a given SIS in performing efficiently its assigned safety function is also measured in terms of SIL. This ability is measured quantitatively according to the SIS operating mode (low demand, high demand and continuous demand), where:

- PFD_{avg} (*the average probability of dangerous failure on demand*): is used for low demand mode and refers to the average unavailability of the SIS.
- PFH (*probability of dangerous failure per hour*): is used for high or continuous demand modes. It represents the average frequency of dangerous failure of the SIS.

The IEC 61508 standard links SIL with PFD_{avg} and PFH as shown in Table 1.

Table 1. Safety integrity levels (SIL) defined according to PFD_{avg} and PFH [1]

SIL	PFD_{avg}	$PFH(h^{-1})$
1	$[10^{-2},10^{-1}]$	$[10^{-6},10^{-5}]$
2	$[10^{-3},10^{-2}]$	$[10^{-7},10^{-6}]$
3	$[10^{-4},10^{-3}]$	$[10^{-8},10^{-7}]$
4	$[10^{-5},10^{-4}]$	$[10^{-9},10^{-8}]$

As technical systems, SIS are exposed to different types of failures that can affect both their ability to appropriately ensure their required safety functions upon demand (safety integrity) and their ability to not activate that functions without a valid demand (operational integrity). Operational integrity refers to the SIS ability in avoiding spurious activations. These failures and their corresponding rates are summarized in Figure 1 [10].

While the safety integrity related performance of SIS is quantitatively measured using PFD_{avg} or PFH, the average probability of failing safely (PFS_{avg}) and the spurious trip rate (STR) are the main quantitative measure of the operational integrity aspect. These measures are practically obtained by summing the performances of the SIS three subsystems, namely sensors (S), logic solver (LS) and final element (FE) as expressed here after [7]:

$$PFD_{avg}^{sis} \approx PFD_S + PFD_{LS} + PFD_{FE} \quad (1)$$

$$PFH_{SIS} \approx PFH_S + PFH_{LS} + PFH_{FE} \quad (2)$$

$$PFS_{avg}^{sis} \approx PFS_S + PFS_{LS} + PFS_{FE} \quad (3)$$

$$STR_{SIS} \approx STR_S + STR_{LS} + STR_{FE} \quad (4)$$

Several contributions have been made to quantify the individual terms in the right-hand side of the above equations using different methods including fault trees, Markov models, Petri nets, analytical expressions, etc. These latter have been the focus in many references starting from the IEC 61508 standard, which provides analytical expressions related to PFD_{avg} and PFH for only many common KooN architectures. The ISA standard [11] offers also expressions for PFD_{avg} and STR for several typical KooN architectures. The Norwegian organization SINTEF [12] provides formulations for PFD_{avg}, PFH, and STR as well as simplified equations for these indicators related to common KooN architectures. A generalization of the PFD_{avg} equations given by ISA (2002) is provided by Oliveira and Abramovitch [13]. We can also find generalized analytical formulations developed by Innal [14] and Dutuit et al. [15] for the four afore mentioned quantitative performance indicators, which are detailed in the study of Innal et al. [7] as described below:

$$PFD_{avg}(KooN) = A_N^{N-K+1}\lambda_{Dind}^{N-K+1}\prod_{i=1}^{N-K+1} MDT_{looi} + \lambda_{DUCCF}.\left(\frac{T_1}{2} + MRT\right) + \lambda_{DDCCF}.MTTR \quad (5)$$

$$PFH(KooN) = A_N^{N-K+1}\lambda_{Dind}^{N-K+1}\prod_{i=1}^{N-K} MDT_{looi} + \lambda_{DUCCF} + \lambda_{DDCCF} \quad (6)$$

$$PFS_{avg}(KooN) \approx A_N^K\,\lambda_{Sind}^K.MDT_{sd}\left[\prod_{i=1}^{K-1} MDTS_{looi}\right] + \left[\beta_{SU}\lambda_{SU} + \beta_{SD}\lambda_{SD}\right].MDT_{SD} \quad (7)$$

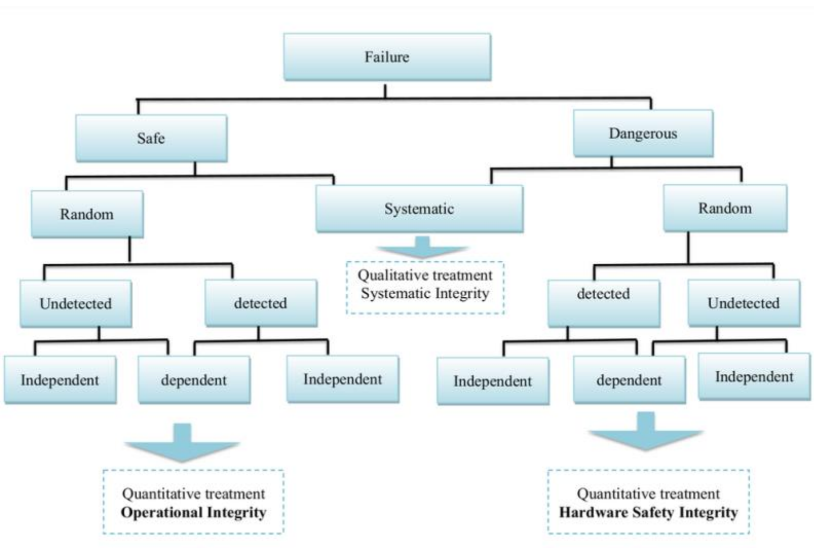


Figure 1. SIS failures classification [10]

$$\mathrm{STR(KooN)} = \mathrm{A_N^K}\,\lambda_{\mathrm{Sind}}^{\mathrm{K}}.\left[\prod_{i=1}^{K-1}\mathrm{MDTS_{looi}}\right] + \left[\beta_{\mathrm{SU}}\lambda_{\mathrm{SU}} + \beta_{\mathrm{SD}}\lambda_{\mathrm{SD}}\right] \tag{8}$$

where:

$$\mathrm{A_N^{N-K+1}} = \frac{\mathrm{N!}}{(\mathrm{K}-1)!} \tag{9}$$

$$\mathrm{MDT_{looi}} = \frac{\lambda_{\mathrm{DUind}}}{\lambda_{\mathrm{Dind}}}.\left(\frac{\mathrm{T_1}}{\mathrm{i+1}} + \mathrm{MRT}\right) + \frac{\lambda_{\mathrm{DDind}}}{\lambda_{\mathrm{Dind}}}.\mathrm{MTTR} \tag{10}$$

$$\mathrm{MDTS_{looi}} = \frac{\lambda_{\mathrm{SUind}}}{\lambda_{\mathrm{Sind}}}.\left(\frac{\mathrm{T_1}}{\mathrm{i+1}} + \mathrm{MRT_S}\right) + \frac{\lambda_{\mathrm{SDind}}}{\lambda_{\mathrm{Sind}}}.\mathrm{MTTR_{SD}} \tag{11}$$

2.2 The life cycle cost

Each project has a life cycle and an underlying cost, called the life cycle cost (LCC). It is defined by the NF EN 60300-3-3 standard [16] as *"The cumulative cost of a product throughout its life cycle"* and by ISO 15663-3 [17] as *"Discounted cumulative total of all costs incurred by a specified function or piece of equipment during its life cycle"*. The life cycle itself is defined by ISO 15663-3 as *"the cycle which includes all stages of development, from the start of the study to the elimination of equipment or a function"*.

The life cycle of safety systems in general is divided into two parts: the construction phase and the operating phase, which include both direct and indirect costs.

Dependability performances (reliability, maintainability and availability in particular) directly influence the cost of a system during its phases of use. For instance, increasing the purchase price often leads to the improvement of the performance of the considered system [16]. The LCC is fundamental for the successful implementation of a safety system and helps make the best choice as well as the optimal allocation of financial resources to achieve the desired objective. The first model specifically developed for process safety systems is based on the subsequent relation [18]:

$$\mathrm{L_{CC}} = \mathrm{L_{AC}} + \mathrm{L_{SC}} + \mathrm{L_{UC}} \tag{12}$$

where, L_{AC} is the life acquisition cost, $\mathrm{L_{SC}}$ is the life support cost, and $\mathrm{L_{UC}}$ is the life unavailability cost.

We also find the model proposed by Goble [19] for safety instrumented systems. It divides the main categories of costs into two parts: supply costs and operating costs. Martorell et al. [20] presented several models for the calculation of the operating cost, taking into account the test and maintenance strategy and also the cost of shutdowns and the cost of overhauling the system. Additionally, Torres-Echeverria [3], and Torres-Echeverria et al. [5] suggested another model to calculate the LCC based on that in the study of Goble [19]. The cost is divided into the cost of supply, operation and risk.

We can also find studies focused on benefit and cost analysis as an interesting method for making decisions related to safety investments, where significant models for calculating costs and benefits are established. In this context and within the framework of process safety, we may cite the study conducted by Reniers and Brijs [21] where the cost was divided into six categories. Moreover, still in the same context, it is worth to mention the approach provided by Chen et al. [22] dedicated to the management of domino effects in chemical industrial areas through a cost-benefit analysis. In addition, a very interesting economic model for allocating safety measures has been developed by Villa et al. [23].

Based on these studies, we developed and adapted the LCC model shown in Table 2 that displays the most important costs related to adding a new safety measure. This model is the basic reference to reach the optimal SIS design at the lowest costs in this study.

In many cases, it is customary to calculate life cycle costs (operating costs) in terms of present value rather than future value. The present value of an annuity is the sum of the present values of all payments. It represents the amount of money that must be invested now in order to make the required future payments. The present value of an annuity can be obtained using the following formula [24], assuming that payments are made at the end of a period, for N payments of M (dinar, dollars, euro, etc.) at a rate-discount from R:

$$\mathrm{PV_A} = \mathrm{M}(1+\mathrm{R})^{-1} + \mathrm{M}(1+\mathrm{R})^{-2} + \ldots + \mathrm{M}(1+\mathrm{R})^{-\mathrm{N}} = \mathrm{M}.\left[1-(1+\mathrm{R})^{-\mathrm{N}}/\mathrm{R}\right] \tag{13}$$

That is why we put notes under the costs of maintenance and examination to pay attention to the distribution of these costs during the life cycle years to give the correct value of the current costs. The same principle is applied for decommissioning costs. If the decommissioning cost was initially agreed upon, which is often the same as the installation cost, we will transfer the cost value to its current value (actualization). However, in the absence of agreement, it would be logical to consider the cost of decommissioning as the cost of future installation (capitalization).

2.3 Cost-effectiveness analysis

Cost-effectiveness analysis (CEA) is a method of analyzing and evaluating projects and it can be seen as a particular form of cost benefit analysis [25]. It is a research method that characterizes the costs of investment related to the amount of benefit that they yield. CEA provides standardized means of comparing investments to identify those that provide maximal effect per incremental unit of cost. Therefore, it is about setting an objective and minimizing the costs to achieve it. For example, one may seek to maximize the safety objective (for instance, the number of lives saved) with a given budget [25]. This optimization problem for determining the optimal combination of safety investments (measures) is similar to solving the so-called knapsack problem [26].

In fact, companies cannot implement all safety measures that are effective or that have passed cost-benefit analysis tests, because they face budgetary constraints, and therefore the choice is linked to the limits of the budget framework. The optimal combination of safety measures canthus be determined through a cost-effectiveness analysis and can be translated into the following mathematical equation, subject to constraints [26]:

$$\begin{cases} \mathrm{Max}\ B_i x_i \\ s.t. \\ C_i x_i \le Bu \\ x_i \in \{0,1\} \end{cases} \tag{14}$$

This equation can be explained as follows [23]. The first term expresses the overall benefit from the portfolio of chosen preventative investments. The second term refers to the first constraint that expresses the overall cost of the chosen measures. It should not be greater than the safety budget (BU).

The last term (the second constraint) expresses a measure either completely taken or not taken at all. Within the limitations of the safety budget, the module's output is the most advantageous combination of safety measures xi for each accident scenario j.

2.4 SIS design optimization

The IEC 61508 standard requires a certain minimum level of safety integrity that should be achieved in the SIS design ph as, in order to reduce the risk to a tolerable level, while satisfying extra objectives that are operational integrity and LCC. Therefore, as stated at the beginning of this section, the best strategy to design an effective SIS is that of trade-off between its safety integrity, operational integrity and the underlying costs throughout its life cycle. More precisely, in light of the developments in this section, the following three objectives should be simultaneously optimized:

- PFD_{avg} (or PFH) which are the basic measures for determining the SIL of the SIS (safety integrity).
- STR that characterizes the number of times the SIS shuts down the protected system unexpectedly and induces production loss (operational integrity). High STR values can lead to loss of confidence in the system. For this reason, as said before, a compromise must be found between PFD_{avg} (or PFH) and STR. One could specify a maximum allowed value for the STR.
- LCC accompanying the achievement of the above desired objectives. Obviously, LCC is considered to consider the budget constraints relating to the SIS life cycle from design to decommissioning.

Hence, SIS design problem is a multi-objective optimization problem in which the goal is to minimize the three above mentioned objectives. Solving this problem requires determining the appropriate values of the decision vector *x* which represents the problem coding.

$$x = [N_S, K_S, S_{type}, S_{T1}, N_{LS}, K_{LS}, LS_{type}, LS_{T1}, N_{FE}, K_{FE}, FE_{type}, FE_{T1}] \quad (15)$$

where, N and K define the KooN architecture specified for each subsystem (S, LS, FE), type refers to the type of component, and T_1 is the proof test interval.

Table 2. Cost calculation model of safety barriers (in particular SIS)

Phase	Cost		Symbol	Calculation
Construction Phase Cost	The design cost		C_{des}	$\sum_{\forall ij} C_{ij}^{des}$ or C_{des}
	The purchase cost Buying price delivery costs		$C_p \begin{cases} C_{Bp} \\ C_d \end{cases}$	$\sum_{\forall ij} C_{ij}^{BP} . N_{ij}$ Loading costs +non-refundable taxes + Unloading costs +other costs.
	The cost of installation The installation price production loss cost		$C_{ins} \begin{cases} C_{ins\,p} \\ C_{p\,l} \end{cases}$	$\sum_{\forall ij} C_{ij}^{ins\,p} . N_{ij}$ $Q.T^{ins}.P$
	The cost of training		C_{TR}	$\sum_{\forall k} C_{Tr.K} . N'_k$
	The start-up cost		$C_{start-up}$	[Q(old)- Q(new)] T' P+ C_{oth}
Operating Phase Cost	Consumption cost		C_{Cn}	$\sum_{\forall ij} Q_{ij} . P_{UNIT}(T - T_{shutdown})$ T=1year = 8630h
	The maintenance cost	Preventive maintenance	C_{PM}	$\sum_{\forall ij} \frac{1}{M_{ij}} . C_{ij}^{PM} . N_{ij}$ Note: pay attention to the interval between maintenance.
		Corrective maintenance	C_{CM}	$\sum_{\forall ij} F_{ij}^{CM} . C_{ij}^{CM} . N_{ij}$ Note: pay attention to the guarantee period.
	The cost of testing		C_T	$\sum_{\forall ij} \frac{1}{TI_{ij}} . C_{ij}^{T} . N_{ij}$ Note: pay attention to the interval between tests.
	Cost of spurious trip		C_{STR}	$STR.C_{SD}$ $C_{SD} = SD_{time}.SD_{loss}$ Each STR causes a system restart so, we have to add the cost of start-up ($C_{start-Up}$)
The Decommissioning Cost	The decommissioning cost		C_{dec}	$C_{ins}(1 + R)^{-N}$ In case of prior agreement C_{dec}= C_{ins}
				$C_{ins}(1 + R)^{N}$ In the absence of a prior agreement about the decommissioning cost
Other costs			C_{oth}	–

i: subsystem subscript; j: technology kind subscript; C_{ij}^{des}: design cost for ij component; C_{ij}^{BP}: bying price for ij component; N_{ij}: number of ij component; $C_{ij}^{ins\,p}$: installation price for ij component; Q: the quantity of hourly production; T^{ins}: installation time (h); P: the product unit price; k: the type of training; $C_{Tr.K}$: the training (k) cost; N'_k: the number of people trained (training k); Q(old): the hourly production quantity before stopping production; Q(new): the quantity of hourly production after stopping production; T': the duration between the moment when the production line is reactivated and time to return to initial production capacity; P: the product unit price; Q_{ij}: the quantity consumed (energy) a unit of time for ij component; P_{UNIT}: the unit price of energy; $T_{shutdown}$: shutdown time; M_{ij}: maintenance frequency of ij component; C_{ij}^{PM}: preventive maintenance cost of ij component; F_{ij}^{CM}: repair frequency of ij components; C_{ij}^{CM}: corrective maintenance (repair) cost of ij component; TI_{ij}: test interval of ij component; C_{ij}^{T}: functional test cost of ij component; STR: spurious trip rate; C_{SD}: cost of shutdown event; SD_{time}: restart time after shutdown; SD_{loss}: cost of loss production per hour; R: rate-discount.

3. USING MANTA RAY FORAGING OPTIMIZATION ALGORITHM TO SOLVE THE SIS DESIGN PROBLEM

The use of evolutionary algorithms (EAs) to solve such multi-objective problem is a common practice in this field.

Genetic algorithms (GA), developed by Holland [27], are one of the most popular meta-heuristics belonging to the class of EAs. They have been extensively used in the context of designing SIS. GA is inspired by the process of natural selection which depends on factors with a biological profile such as: Mutation, crossover and selection. Obviously, the main goal of GA is to find the optimal configuration for a given optimization problem by applying a good balance between exploitation and exploration of the search space. Detailed descriptions of GA can be found in the study of Gen et al. [28], Katoch et al. [29], Bendine [30], Fonseca and Fleming [31]. Figure 2 summarizes the main steps of GA algorithm.

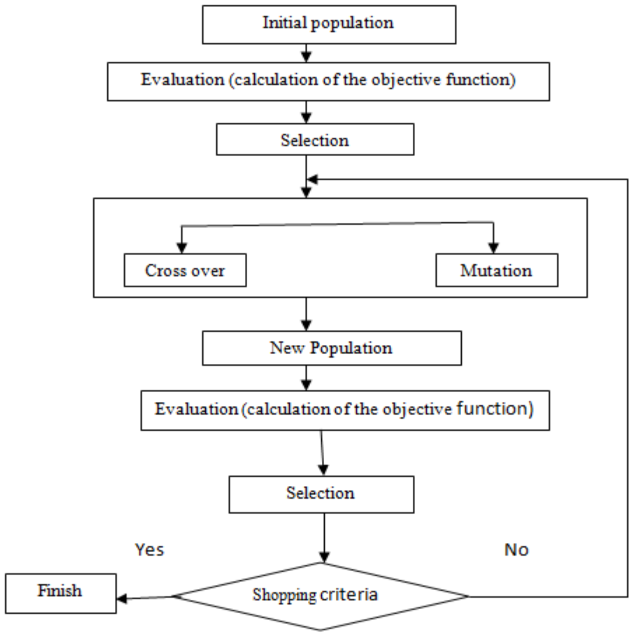


Figure 2. Flowchart of a genetic algorithm [30]

In the last few decades, many other prominent algorithms have been developed to deal with different complex real-world problems. Under this context, we can quote the so-called MRFO, which represents a novel bio-inspired optimization approach developed by Zhao et al. [32] in 2019. MRFO is a meta-heuristic belonging to the class of swarm intelligence algorithm. Studies and comparisons have shown that this approach is often superior to other well-known algorithms [32]. MRFO presents a strong global optimization ability on both constrained and unconstrained problems and it is very suitable for handling real-world problems, including SIS design problem.

Manta Ray is one of the largest known marine creatures belonging to the genus *Mobula*. They are classified among the *Myliobatiformes* and are placed in the family *Myliobatidae*. They have the largest brains and brain to body ration of all fish. The average life span of these fish is 20 years [33]. These fish attract attention and interest due to their ability to find plankton whatever the circumstances (a grown-up manta ray can eat 5 kg of plankton on everyday), this is due to its unique and clever foraging strategy. That's why it inspired researchers to create a new optimization method simulates the cooperative behavior observed in manta ray to provide food. These fish rely on many strategies to search for food, which are: straight, surface, chain, piggy-back, bottom, and sideways [34]. But MRFO algorithm simulates the following three methods foraging: chain, cyclone, and somersault [32], which can be described as follows:

✓ *Chain Foraging*

In this strategy, a group of manta rays move in the form of an organized line, lining up one behind the other, they travel forward and backward their fins open in front of their mouth [35]. We also notice in this movement the support of the smaller male manta rays by the females, by swimming over their back bellies [36]. The first manta ray updates its location (current position) based on the best solutions obtained so far, while the rest of the manta ray updates its current position according to the best solution and the location of the manta ray in front of it in the search area. This can be translated by the following equation [9, 32]:

$$x_i^{t+1} = \begin{cases} x_i^t + r(G_{best}^t - x_i^t) + 2.r.\sqrt{|\log(r)|}\,.(G_{best}^t - x_i^t) & i = 1 \\ x_i^t + r(x_{i-1}^t - x_i^t) + 2.r.\sqrt{|\log(r)|}\,.(G_{best}^t - x_i^t) & i = 2, \dots N \end{cases} \tag{16}$$

where, r is a random vector in [0, 1], N is the size of population, x_i^t is the position or the ith manta ray in the iteration t and x_i^{t+1} is its new position in the next iteration, and G_{best} represent the global best solution within the entire population.

✓ *Cyclone Foraging*

This strategy is used in places rich in food, where dozens of manta ray fish gather to form a spiral. This circle's diameter is proportional to the number of manta rays (approximately 15-20 m), and this cyclone always rotates and clockwise this is to create a current that attracts prey outside the feeding circle towards them [35]. To simulate this motion, a spiral equation is used to update the position of the population [9, 32]:

$$x_i^{t+1} = \begin{cases} G_{best} + r.\left(G_{best}^t - x_i^t\right) + 2e^{r_1\frac{T_{max}-t+1}{T_{max}}}.\sin(2\pi r_1).\left(G_{best}^t - x_i^t\right) & i = 1 \\ G_{best} + r.\left(x_{i-1}^t - x_i^t\right) + 2e^{r_1\frac{T_{max}-t+1}{T_{max}}}.\sin(2\pi r_1).\left(G_{best}^t - x_i^t\right) & i = 2, \dots, N \end{cases} \tag{17}$$

where: T_{max} is the maximum number of iterations and r_1 is a random number in [0, 1]. In order to improve the exploratory ability, each individual updates his position away from the current best position and according to a new random position in the entire search space as follows [8, 32]:

$$x_i^{t+1} = \begin{cases} x_{rand} + r.\left(x_{rand}^t - x_i^t\right) + 2e^{r_1\frac{T_{max}-t+1}{T_{max}}}.\sin(2\pi r_1).\left(x_{rand}^t - x_i^t\right) & i = 1 \\ x_{rand} + r.\left(x_{i-1}^t - x_i^t\right) + 2e^{r_1\frac{T_{max}-t+1}{T_{max}}}.\sin(2\pi r_1).\left(x_{rand}^t - x_i^t\right) & i = 2, \dots, N \end{cases} \tag{18}$$

where, x_{rand} is a random reference point in the search space given by:

$$x_{rand} = LB + r.(UB - LB) \tag{19}$$

LB: lower boundary of the search space.
UB: upper boundary of the search space.

✓ *Somersault Foraging*

This strategy of feeding is typically used when the prey is concentrated near the surface to limit mobility and improve feeding effectiveness [35]. The manta ray performs a series of backwards somersaults, which are random, repetitive, local and cyclical movements, and it is one of the most beautiful scenes in nature [32]. In this strategy, the manta ray update their position around the best position found so far by performing a somersault movements. Therefore, its mathematical model is given by [9, 32]:

$$x_i^{t+1} = x_i^t + S.(r_2.G_{best} - r_3.x_i^t), i = 1, \dots, N \tag{20}$$

S is the somersault factor that defines the somersault range of manta rays and it is set to 2. r_2 and r_3 are random numbers between 0 and 1.

Based on the above description of MRFO algorithm, it is clear that there is a big difference between MRFO and GA algorithms. Indeed, and according to our point of view, the main difference between them is in the manner of how they deal with exploration and exploitation strategies during the optimization process. Hence, GA ensures the exploration by applying crossover operators, and exploitation by applying mutation operators, while MRFO performs some random movement to ensure the exploration, and it performs some oriented movements by following the current global best position at the hope of exploiting the promising regions in the search space.

The MRFO [32] starts by creating a random population in the domain of the problem, after this step, each individual updates its position on each iteration with relation to the individual in front of it as well as the reference position. The change in the value of t/T allows exploratory and exploitative research to be conducted: for $t/T < rand$ the current best solution is selected as the exploitation reference position, and for $t/T > rand$ is selected as a reference position for exploration. And according to the value of rand, the MRFO can switch between the two strategies chain foraging and cyclone foraging. Then, by foraging somersaults the individuals update their positions in relation to the best position found so far. These operations and calculations are done interactively and stop when the specified stop conditions are met. Finally, the fitness value and the position of the best individual are returned.

To solve the multi-objective SIS problem, we use Multi-Objective Manta Ray Foraging Optimizer (MOMRFO) [9]. This algorithm uses an external archive to maintain historical record of Pareto solutions by storing the non-dominant solutions obtained so far. However, and for runtime reasons, this archive should be limited to a given maximum size (T_max). Hence, it will be carefully updated during the optimization process to identify the solutions that will be accepted to be stored, and those that are not accepted (because the archive is limited). Moreover, the archiving strategy should maintain a good balance between convergence and diversity of solutions in the search space. For this reason, the MOMRFO algorithm adopts an effective archiving strategy based on the grid adaptive mechanism. This technique consists of dividing the external archive into a certain number of hyper cubes containing a certain number of solutions. So, the number of solutions in each hypercube represents the density of this hypercube, and this density helps to identify the most and the less crowded regions in the archive. Accordingly, if the archive is full, the removed solutions will be removed from the high crowded regions, and when a new solution is added, it will be added in the less crowded regions. The MOMRFO algorithm also depends on the way of choosing the Global best solution global, knowing that this solution guides the population towards well-distributed regions in the Pareto front. For this reason, a roulette wheel is used to identify the area that may contain probably these solutions for improving both convergence and diversity. Finally, the algorithm returns the final archive containing the resulting Pareto front.

It is worth mentioning that the computational complexity of MRFO algorithm is of O(TN), where T is the maximum number of iterations, and N is the number of individuals. On the other hand, the complexity of the update archive procedure is of $O(N^2)$. Accordingly, the complexity of MOMRFO can be estimated by $O(N^2)$. This complexity is similar to that of the selected GA algorithm.

4. APPLICATION EXAMPLE

The widespread use and applications of SIS operating in low demand mode is evident across a variety of industrial sectors. The general form of processing such usage is almost the same despite in the involved diversity in the measured parameters, the provided functions and the nature of the applications themselves. To highlight the utility of the discussed algorithm, we take as a basis a simple example treated by Innal et al. [7] of designing a SIS operating in a low demand mode. Obviously, the realization of the optimal SIS requires the consideration of several design options since the optimization is centered on the variability of the redundancy and the diversity of the SIS subsystems elements. Supposing that a SIL 3 is required, the value of PFD_{avg} of the entire SIS will be constrained as follows: $PFD_{avg}^{sis} \leq PFD_{avg}^{max} = 1E-3$. Therefore, this multi-objective problem with constraints takes the form:

$$\begin{cases} Y = F(X) = \left(PFD_{avg}(X); STR(X); LCC(X)\right) \\ PFD_{avg} \leq 10^{-3} \\ K_1 \leq N_1;\ K_2 \leq N_2;\ K_3 \leq N_3 \end{cases} \tag{21}$$

The employed data in the original application by Innal et al. [7] in addition to some supplementary factors are shown in Table 3.

4.1 Using genetic algorithms to solve the problem

At this level, we follow the conventional method of solving the SIS design problem using GA. For this we use the GA-based solver in the optimization toolbox in MATLAB [37]. For this, we take the following parameters: population size (150), selection type (Tournament), crossover function (Two points), crossover fraction (0.8), mutation function (Adaptive feasible), the stopping criterion (maximum number of generations=200). Setting these values is performed by testing different possible alternatives focusing on the reasonable combination of the computation time and the fitness levels.

Some of the obtained non-dominated solutions with their relative values for the three evaluated objectives are also included (Pareto front) are given in Table 4. Additionally, Figure 3 shows the Pareto front related to the various studied objectives given in a 2D presentation (PFD_{avg} and STR; PFD_{avg} and cost; STR and cost) and in a 3D presentation (PFD_{avg}, STR and cost).

All of the resulting solutions represent optimal SIS systems, and the choice between them will be in the hands of decision makers based on personal preferences, values, and trade-offs in relation to the objectives being examined.

4.2 Using the Manta Ray foraging algorithm to solve the problem

At this level, MRFO is used to solve SIS design problem. Since the dimension of SIS problem includes 12 parameters, each manta ray is defined in 12-dimensional search space (12 positions) so as each dimension refers to a given parameter each position represented a decision variable. The positions of this Manta Ray take variable values between the lower limits: [1 1 1 1 1 1 1 1 1 1 1 1], and the upper limits: [55 3 4 3 3 3 3 4 4 3 4].

To achieve KooN vote that represents the linear inequality constraints we use the static penalty method for guide the search to feasible regions, by adding a penal value in the objective function as follows [38]:

$$f_m(x) = f_m(x) + \sum_{i=1}^{p} P_i . \max(g_i(x), 0) + \sum_{i=1}^{K} P_i . \max(|h_i(x)| - \delta, 0) \qquad (22)$$

where:

$f_m(x)$, m=1, 2, … M are the objective function to be optimized.

$G_i(x) \le 0$, i=1, 2… P are inequality constraints.

$H_i(x) = 0$, i=1, 2 … K are equality constraints.

P_i and δ are respectively the penalty factor and the tolerance on the equality constraints to consider it as feasible.

For constraints on the objective function $PFD \le 10^{-3}$, it will be achieved by rejecting solutions that do not meet this condition from the external archive of MOMRFO during the optimization process. The used parameters are: population size (150), Maximum Number of Iterations (200), the maximum size of archive (100).

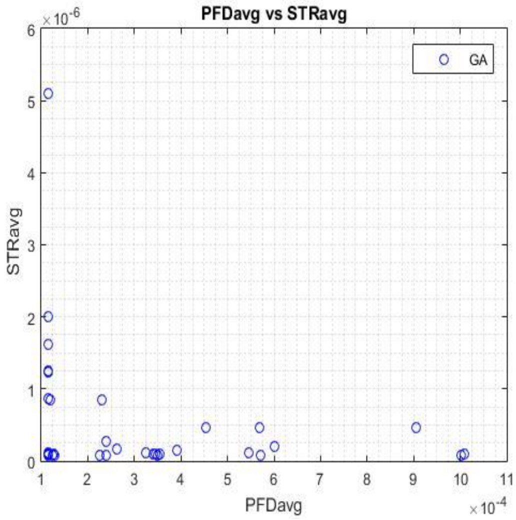
Some of the obtained results are given in Table 5, while the visual presentation of the obtained solutions is given in Figure 4.

Table 3. Input data

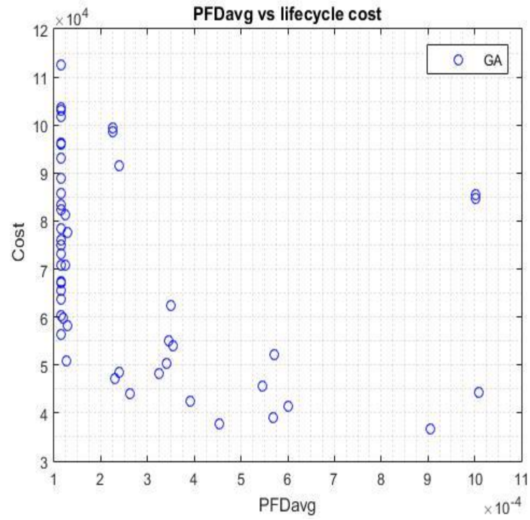
Data	**Types of Components: λ 10-6(h); MTTR(h); C_P(u);C_T(u);$\beta_{DU}=\beta=\beta_{SU}=2\beta_D=2\beta_{SD}$**			**T1(h)**
Sub-systems	**Type 1**	**Type 2**	**Type 3**	
PT N1Max=5	λ_D=0.151	λ_D=1.9	λ_D=4.11	
	DC=0.318	DC=0.51	DC=0.1	
	λ_S=0.383	λ_S=2.16	λ_S=6.81	4380
	DC_S=0.692	DC_S=0.56	DC_S=0.1	8760
	$\beta = 0.02$	$\beta = 0.02$	$\beta = 0.02$	13140
	$MTTR_{DD}$=4	$MTTR_{DD}$=8	$MTTR_{DD}$=10	17520
	$MTTR_{SD}$=8	$MTTR_{SD}$=10	$MTTR_{SD}$=10	
	C_P=4844	C_P =2306	C_P =500	
	C_T=60	C_T =30	C_T =20	
LS N2Max=3	λ_D=0.01	λ_D=10	λ_D=15	
	DC=0.9	DC=0.9	DC=0.67	
	λ_S=0.01	λ_S=10	λ_S=15	
	DC_S =0.2	DC_S =0.2	DC_S =0.2	8760
	$\beta = 0.01$	$\beta = 0.01$	$\beta = 0.01$	13140
	$MTTR_{DD}$ =4	$MTTR_{DD}$ =8	$MTTR_{DD}$ =8	17520
	$MTTR_{SD}$ =4	$MTTR_{SD}$ =8	$MTTR_{SD}$ =10	
	C_P=4000	C_P=2800	C_P=2000	
	C_T =70	C_T =50	C_T =40	
SDV N3Max=4	λ_D=3.35	λ_D=5.44	λ_D=7.9	
	DC=0.25	DC=0.20	DC=0.1	
	λ_S=3.94	λ_S=3.17	λ_S=9.17	4380
	DC_S =0	DC_S =0	DC_S =0	8760
	$\beta = 0.02$	$\beta = 0.05$	$\beta = 0.1$	13140
	$MTTR_{DD}$ =8	$MTTR_{DD}$ =8	$MTTR_{DD}$ =10	17520
	$MTTR_{SD}$ =8	$MTTR_{SD}$ =10	$MTTR_{SD}$ =15	
	C_P =6940	C_P =6500	C_P =6000	
	C_T =90	C_T =70	C_T =60	
Design/install/commissioning PLC=500(u)				
Repair PLC =500 (u/event)				
Shut down time =24(h)				
Maintenance PLC=150 (u/event)				
Design overall instrumentation =2000 (u)				
Installation/commissioning per instrument =300 (u)				
Maintenance per instrument =70 (u/event)				
Repair cost per instrument & PLC = 200 (u/event)				
Cost loss of production =2000 (u/h)				
SIS life =15 (years)				
R=4%				
guarantee period=1year for each component				

Table 4. Some selected solution using GA

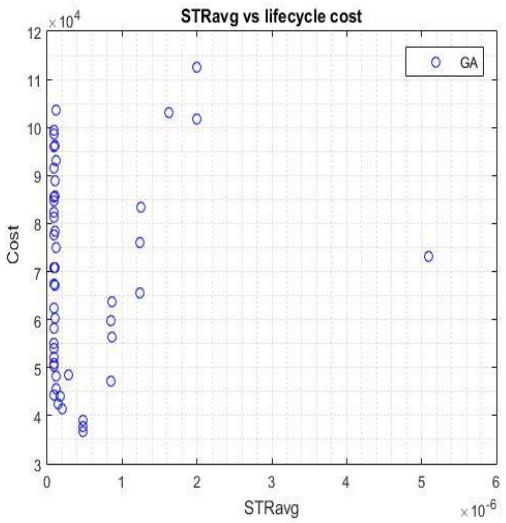
No.	Variables												Objective		
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	PFD_{avg}^{sis}	STR_{avg}^{sis} (h^{-1})	Cost(u)
01	5	1	1	1	3	1	1	1	4	3	1	1	1,1508E-4	2,0036E-6	1,1240E+5
02	3	2	1	1	3	2	1	1	3	3	1	1	1,1530E-4	8,4481E-8	9,5993E+4
03	1	1	1	2	1	1	1	2	2	2	3	3	9,2623E-4	4,8470E-7	3,4059E+4
04	1	1	1	1	1	1	1	1	2	2	1	1	3,4108E-4	4,7180E-7	3,9948E+4
05	4	2	1	1	1	1	1	1	3	2	1	1	1,1946E-4	9,4972E-8	7,1013E+4
06	1	1	1	1	1	1	1	1	2	2	1	2	4,5114E-4	4,7180E-7	3,7945E+4
07	1	1	1	2	1	1	1	2	2	2	1	4	8,9898E-4	4,7180E-7	3,5648E+4
08	5	2	1	1	3	1	1	1	4	2	1	1	1,1508E-4	1,1557E-7	1,0357E+5
09	2	1	1	3	1	1	1	2	2	2	1	1	1,3127E-4	8,4979E-7	4,7080E+4
10	3	1	1	1	3	2	1	1	2	2	1	1	1,1509E-4	1,2179E-6	7,2797E+4
11	2	2	1	1	2	2	1	1	2	2	1	1	6,7713E-4	8,4093E-8	5,0814E+4
12	2	2	1	1	3	2	1	1	4	3	1	1	5,5831E-4	8,4093E-8	8,1123E+4
13	5	2	1	1	2	1	1	1	4	2	1	1	1,1508E-4	1,0566E-7	9,6274E+4
14	4	3	1	1	1	1	1	1	4	3	1	1	3,8923E-4	4,8470E-7	3,6523E+4
15	1	1	1	1	1	1	1	2	2	2	3	1	5,6706E-4	8,4093E-8	7,3874E+4
16	2	2	1	1	2	2	1	1	4	3	1	1	1,1510E-4	8,5062E-8	8,5463E+4
17	4	2	1	1	3	2	1	1	3	3	1	1	7,5232E-4	4,7180E-7	4,2395E+4
18	3	2	2	3	1	1	1	2	2	2	2	1	4,9495E-4	1,7466E-7	4,9383E+4
19	4	2	1	1	2	2	1	2	2	2	1	1	1,2820E-4	8,5062E-8	6,7146E+4



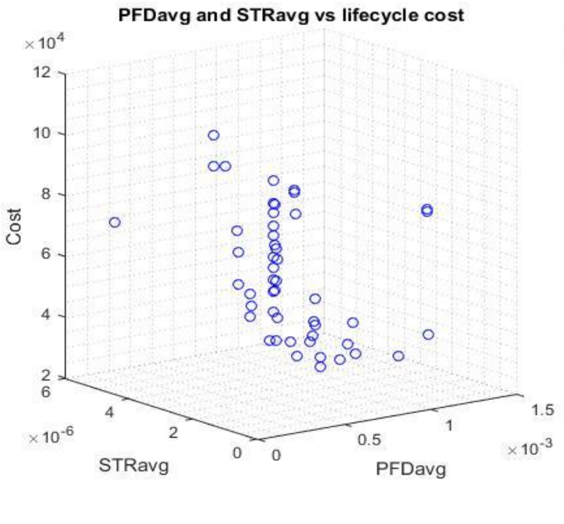
(a) PFD_{avg} vs STR_{avg} (GA)



(b) PFD_{avg} vs LCC(GA)



(c) STR_{avg} vs LCC(GA)

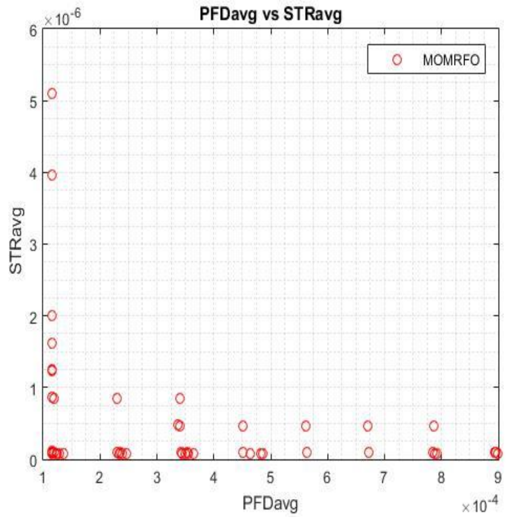


(d) PFD_{avg} and STR_{avg} vs LCC(GA)

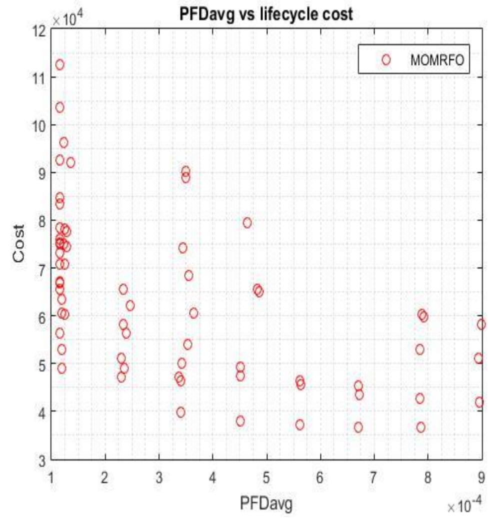
Figure 3. Obtained Pareto solutions using GA

Table 5. Some selected solutions using MRFO

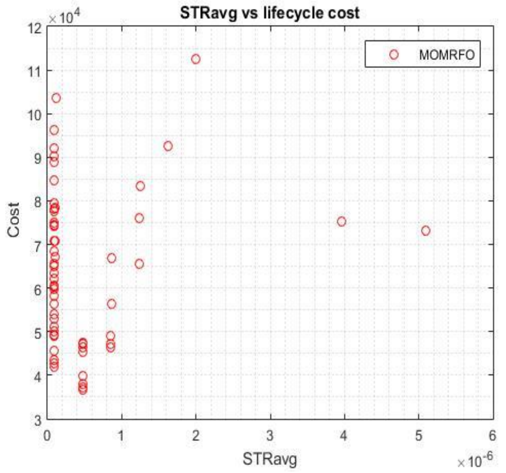
No.	Variables												Objective		
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	PFD_{avg}^{sis}	STR_{avg}^{sis} (h^{-1})	Cost(u)
01	3	1	1	1	3	1	1	1	3	2	1	1	1,1508E-4	1,2475E-6	8,3465E+4
02	3	2	1	2	1	1	1	1	2	2	1	2	2,3482E-4	9,4965E-8	4,9048E+4
03	2	1	1	1	1	1	1	1	2	2	1	1	1,1953E-4	8,4979E-7	4,9146E+4
04	3	3	1	1	2	2	1	1	2	2	1	2	8,9875E-4	8,3900E-8	5,8246E+4
05	5	4	1	2	3	2	1	3	4	4	1	1	1,2227E-4	8,3900E-8	9,8498E+4
06	1	1	1	1	1	1	1	1	2	2	1	3	5,6120E-4	4,7180E-7	3,7252E+4
07	3	2	1	1	2	2	1	1	3	3	1	1	1,2405E-4	8,4481E-8	7,0779E+4
08	5	4	1	1	3	3	1	1	3	3	1	3	3,4901E-4	8,3900E-8	8,8849E+4
09	3	2	1	1	2	1	1	1	3	2	1	1	1,1528E-4	1,0430E-7	7,0872E+4
10	4	3	1	1	1	1	1	1	2	2	1	2	2,2992E-4	9,3810E-8	5,8477E+4
11	1	1	1	1	1	1	1	1	2	2	1	2	4,5114E-4	4,7180E-7	3,7945E+4
12	3	3	1	1	2	2	1	1	2	2	1	1	7,8869E-4	8,3900E-8	6,0247E+4
13	4	3	1	1	1	1	1	1	2	2	1	1	1,1986E-4	9,3810E-8	6,0478E+4
14	4	3	1	1	2	2	1	2	3	3	1	1	1,2860E-4	8,3900E-8	7,7670E+4
15	2	2	1	1	1	1	1	1	2	2	1	2	6,7276E-4	9,4003E-8	7,3612E+4
16	3	3	1	1	2	2	1	1	2	2	1	2	8,9850E-4	8,3900E-8	5,8246E+4
17	2	1	1	1	2	1	1	1	1	1	1	1	1,1515E-4	4,7209E-6	6,3963E+4
18	2	2	1	1	1	1	1	2	2	2	1	4	8,9507E-4	9,4003E-8	4,1982E+4
19	5	3	1	1	2	1	1	1	4	4	1	1	1,2386E-4	8,3901E-8	9,6172E+4



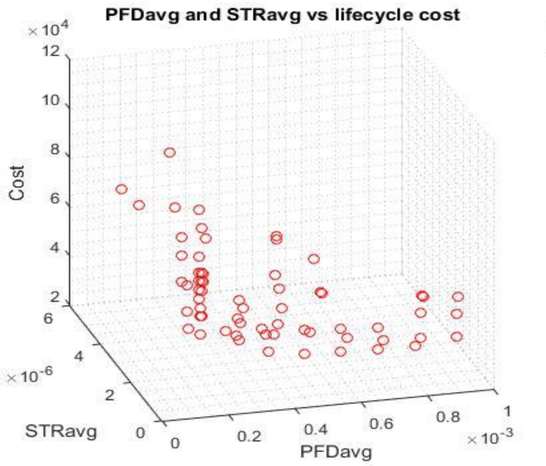
(a) PFD_{avg} vs STR_{avg} (MRFO)



(b) PFD_{avg} vs LCC(MRFO)



(c) STR_{avg} vs LCC(MRFO)



(d) PFD_{avg} and STR_{avg} vs LCC(MRFO)

Figure 4. Obtained Pareto solutions using MRFO

The same observation, all these solutions are ideal systems and the choice remains in the hands of the decision maker. Also, through visual observation only on Figures 3 and 4, we notice that MRFO is preferred compared to GA in terms of number of solutions extracted by each algorithm. Indeed, this perspective is relatively important when solving SIS design problem and it gives to the decision maker more options regarding the SIS design. Hence, it can be seen that the GA has provided 54 solutions, while the MOMRFO algorithm has provided 100 solutions. That is to say, the MOMRFO can offer more alternatives and a large wide of choices that can satisfy as well as possible the preferences of the decision-maker.

To summarize, this paper's focus is on extending the level of detail in the formulation of the SIS design problem, especially regarding the life cycle cost with what it holds of complexity. On the other hand, it has been shown that substituting the traditional practice of relying on GA by recent alternatives (mainly MRFO in this case) can provide many practical benefits. This includes primarily enriching the decision maker's range of choice.

5. CONCLUSIONS

The critical and complex nature of SIS necessitates the appropriate handling of its design. This is important to ensure that introducing this solution will be beneficial in all respects. This mainly includes the ability of the SIS to perform its assigned safety function and to cause no or acceptable levels of disruption. In addition to these two aspects that are presented in Subsection 2.1, it is also crucial to ensure that SIS is aligned with the overall resources allocation strategy and objectives as discussed in Subsection 2.2.

In this paper, a new cost-effective analysis framework is proposed to handle the SIS designing problem. The focus at the first level is on the detailed study of the involved costs with their classification and practical estimation. On the side, the objective is the employment of an efficient algorithm that can facilitates the handling of such a complex problem. For this end, the recently developed MRFO algorithm is considered to solve the multi-objective SIS design problem in comparison with the common use of GA. The obtained results confirmed the superiority of the former algorithm in terms of the number of the extracted solutions, therefore the number of options granted to the decision maker. Consequently, the practical benefits of this proposed framework lie in improving the accuracy of overall model through the detailed consideration of the involved costs in addition to the ability of the employed algorithm to explore wider regions to provide richer choice ranges. Such findings highlight the importance of revising the current practices in dealing with these kinds of problems to enhance their practical utility.

The proposed framework this paper is dedicated to handle the problem of designing SIS that reduce the risk level to a certain predefined tolerable level. The objective for future works is to extend it to the case when the use of SIS is controlled also by its practicability. Such an extension involves many challenges regarding the formulation of the whole problem in addition to the complexity and diversity of the associated parameters. Another objective is to conduct a detailed study to develop a clear view on the criteria and conditions needed for the proper handling of the real-world SIS design problem with its different facets.

REFERENCES

[1] IEC 61508. (2010). Functional Safety of Electrical/electronic/programmable Electronic Safety Related Systems, 2nd ed., IEC 61508 Standard, Geneva. https://webstore.iec.ch/publication/5515, accessed on Apr. 28, 2023.

[2] IEC 61511. (2016). Functional Safety – Safety Instrumented Systems for the Process Industry Sector. In: 2nd ed., International Standard, Geneva.

[3] Torres-Echeverria, A.C. (2009). Modeling and optimization of safety instrumented systems based on dependability and cost measures. Ph.D. Dissertation, Department of Electronic Engineering, The University of Sheffield, USA.

[4] Torres-Echeverria, A.C., Martorell, S., Thompson, H.A. (2009). Modeling and optimization of proof testing policies for safety instrumented systems. Reliability Engineering and System Safety, 94(4): 838-854. https://doi.org/10.1016/j.ress.2008.09.006

[5] Torres-Echeverria, A.C., Martorell, S., Thompson, H.A. (2009). Design optimization of a safety instrumented system based on RAMS+C addressing IEC 61508 requirements and diverse redundancy. Reliability Engineering and System Safety, 94(2): 162-179. https://doi.org/10.1016/j.ress.2008.02.010

[6] Torres-Echeverria, A.C., Martorell, S., Thompson, H.A. (2012). Multi-objective optimization of design and testing of safety instrumented systems with MooN voting architectures using a genetic algorithm. Reliability Engineering and System Safety, 106: 45-60. https://doi.org/10.1016/j.ress.2012.03.010

[7] Innal, F., Dutuit, Y., Chebila, M. (2015). Safety and operational integrity evaluation and design optimization of safety instrumented systems. Reliability Engineering and System Safety, 134: 32-50. https://doi.org/10.1016/j.ress.2014.10.001

[8] Touahar, H., Ouazraoui, N., Khanfri, N.E.H., Korichi, M., Bachi, B., Boukrouma, H.E. (2021). Multi-objective optimization of safety instrumented systems maintenance strategy: A case study. International Journal of Quality & Reliability Management, 38(8): 1792-1815. https://doi.org/10.1108/IJQRM-03-2020-0076

[9] Got, A., Zouache, D., Moussaoui, A. (2022). MOMRFO: Multi-objective Manta ray foraging optimizer for handling engineering design problems. Knowledge-Based Systems, 237: 107880. https://doi.org/10.1016/j.knosys.2021.107880

[10] Chebila, M. (2015). Modélisation et evaluation des performances des systemes instrumentes de securité avec prise en compte des incertitudes. Ph.D. Dissertation, University of Batna 2.

[11] ISA-TR84.00.02-2002. (2002). Safety instrumented functions (SIF)–safety integrity level (SIL) evaluation techniques, Part 2: Determining the SIL of A SIF via simplified equations. Instrumentation, Systems and Automation Society (ISA). North Carolina, USA.

[12] Hauge, S., Lundteigen, M.A., Hokstad, P., Håbrekke, S. (2010). Reliability prediction method for safety instrumented systems–pds method handbook. 2010 edition. SINTEF report STF50 A, 6031, 460.

[13] Oliveira, L.F., Abramovitch, R.N. (2010). Extension of ISA TR84. 00.02 PFD equations to *KooN* architectures. Reliability Engineering & System Safety, 95(7): 707-715. https://doi.org/10.1016/j.ress.2010.01.006

[14] Innal, F. (2008). Contribution to the modeling of safety instrumented systems and the evaluation of their performances: Critical analysis of the IEC 61508 standard. Doctoral Dissertation, Bordeaux1.

[15] Dutuit, Y., Innal, F., Deconinck, G. (2009). Etude Complémentaire des Systèmes Instrumentés de Sécurité.

Total (French Oil Company) Report.
[16] IEC 60300-3-3. (2004). Dependability management – Part 3-3: Application Guide – Life cycle Costing.
[17] ISO 15663-3. (2001). Petroleum and natural gas industries – Life cycle costing – Part 3: Implementation guidelines.
[18] Lydersen, S., Aaroe, R. (1989). Life cycle cost prediction handbook, computer-based process safety systems. SINTEF Safety and Reliability.
[19] Goble, W.M. (1998). Control Systems Safety Evaluation and Reliability. ISA, ISBN# 1-55617-636-8. Reference on FMEDA Methods Failure Rate Calculations, 3.
[20] Martorell, S., Sánchez, A., Carlos, S., Serradell, V. (2002). Simultaneous and multi-criteria optimization of TS requirements and maintenance at NPPs. Annals of Nuclear Energy, 29(2): 147-168. https://doi.org/10.1016/S0306-4549(01)00037-8
[21] Reniers, G.L.L., Brijs, T. (2014). An overview of cost-benefit models/tools for investigating occupational accidents. Chemical Engineering Transactions, 36: 43-48. https://doi.org/10.3303/CET1436008
[22] Chen, C., Reniers, G., Khakzad, N. (2020). Cost-benefit management of intentional domino effects in chemical industrial areas. Process Safety and Environmental Protection, 134: 392-405. https://doi.org/10.1016/j.psep.2019.10.007
[23] Villa, V., Reniers, G.L., Paltrinieri, N., Cozzani, V. (2017). Development of an economic model for counter terrorism measures in the process-industry. Journal of Loss Prevention in the Process Industries, 49: 437-460. https://doi.org/10.1016/j.jlp.2017.06.001
[24] Piermay, M., Hereil, O., Lazimi, A. (1989). Mathématiques Financières. Pearson Education France.
[25] Treich, N. (2005). L'analyse coût-bénéfice de la prévention des risques. LERNA-INRA, Universityof Toulouse.
[26] Reniers, G., Sörensen, K. (2013). Optimal allocation of safety and security resources. Chemical Engineering Transactions, 31: 397-402. https://doi.org/10.3303/CET1331067
[27] Holland, J. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor.
[28] Gen, M., Cheng, R., Lin, L. (2008). Network models and optimization: Multiobjective genetic algorithm approach. Springer Science & Business Media. https://doi.org/10.1007/978-1-84800-181-7
[29] Katoch, S., Chauhan, S.S., Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. Multimedia Tools and Applications, 80: 8091-8126. https://doi.org/10.1007/s11042-020-10139-6
[30] Bendine, K. (2016). Contribution to the study of active control of composite structures by incorporating intelligent materials. Ph.D. Dissertation. University SidiBel Abbes.
[31] Fonseca, C.M., Fleming, P.J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. Icga, 93: 416-423.
[32] Zhao, W., Zhang, Z., Wang, L. (2020). Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. Engineering Applications of Artificial Intelligence, 87: 103300. https://doi.org/10.1016/j.engappai.2019.103300
[33] Bancroft, E.N. (1829). On the Fish Known in Jamaica as the Sea-Devil. W. Phillips, USA.
[34] Hemeida, M.G., Alkhalaf, S., Mohamed, A.A.A., Ibrahim, A.A., Senjyu, T. (2020). Distributed generators optimization based on multi-objective functions using manta rays foraging optimization algorithm (MRFO). Energies, 13(15): 3847. https://doi.org/10.3390/en13153847
[35] Stevens, G.M.W. (2016). Conservation and population ecology of manta rays in the Maldives. Ph.D. Dissertation, University of York, USA.
[36] Rizzo, J. (2016). Ocean Animals: Who's Who in the Deep Blue. National Geographic Books.
[37] MATLAB, R2009 version. The MathWorks, Inc: www.mathworks.com.
[38] Rao, S.S. (2019). Engineering Optimization: Theory and Practice. John Wiley & Sons.

NOMENCLATURE

A_n^k	Number of arrangements of size k from a set with n elements
C_p	Purchase cost
C_T	Proof tests cost
DC	Diagnostic coverage for dangerous failures
DC_s	Diagnostic coverage for safe failures
FE	Final elements
LS	Logic solver
$MDTS_{1ooi}$	Mean down time for 1ooi architecture due to independent safe failures
MDT_{sd}	Mean down time consecutive to a shutdown
MRT	Mean repair time for DU failures
MRT_s	Mean repair time for SU failures
MTTR	Mean time to restoration for DD failures
$MTTR_{SD}$	Mean time to restoration for SD failures
PFD_{avg}	Average probability of failure on demand
PFD_{avg}^{SIS}	SIS average PFD
PFD_{avg}^{max}	Maximum allowed value for PFD_{avg}^{SIS}
PFD_{KooN}	PFD for KooN architecture
PFH_{SIS}	SIS probability of dangerous failure per hour (average)
PFH_{KooN}	PFH for KooN architecture
S	Sensor
STR_{SIS}	SIS spurious trip rate (average)
STR_{KooN}	STR for KooN architecture
T_1	Proof tests interval
β	CCF proportion (β factor)
R	Discount rate
N	Years life for the system
β_{DU}	β for dangerous undetected (DU) failures
β_{DD}	β for dangerous detected (DD) failures
β_{SD}	β for safe detected (SD) failures
β_{SU}	β for safe undetected (SU) failures
λ_D	Dangerous failure rate
λ_{Dind}	Independent dangerous failure rate
λ_{DCCF}	Dependent dangerous failure rate (CCF)
λ_{DD}	DD failure rate
λ_{DDind}	Independent DD failure rate
λ_{DDCCF}	Dependent DD failure rate
λ_{DU}	DU failure rate
λ_{DUind}	Independent DU failure rate
λ_{DUCCF}	Dependent DU failure rate