
An intelligent novel tripartite - (PSO-GA-SA) optimisation strategy

Kayode Owa* and Lisa Jackson

Department of Aeronautical and Automotive Engineering,
Loughborough University,
Leicestershire LE11 3TU, UK
Email: k.o.owa@lboro.ac.uk; olayemi_owa@yahoo.co.uk
Email: l.m.jackson@lboro.ac.uk
*Corresponding author

Tom Jackson

Centre for Information Management,
The School of Business and Economics,
Loughborough University,
Leicestershire LE11 3TU, UK
Email: t.w.jackson@lboro.ac.uk

Abstract: This paper presents a tripartite version of particle swarm optimisation, genetic algorithm, and simulated annealing (PSO-GA-SA) optimisation strategy addressing some predominant issues such as the problem of the potential solution being trapped in a local minima solution space, the untimely convergence and the slow rate of arriving at optimal solutions. This strategy is designed with an intelligence beneficiary trade-off between exploration and exploitation of the full potential of all the capabilities of PSO, GA, and SA functioning simultaneously. The design algorithm further incorporates a variable velocity component that introduces random intelligence. There are substantial performance improvements when the novel robust design is first validated with three test functions for the initial case studies. To demonstrate the capabilities to handle complexities and establish scalability in the implementation of the proposed approach, the optimisation strategy is further applied to a high-integrity protection system (HIPS) which is a real-life safety system design optimisation problem with increased number of input variables, constraints, and limitations on the available resources. The novel design performs better than their individual methods using the number of fitness evaluations, as the performance metrics, whilst operating with both a reduced number of generations and initial number of starting potential solutions.

Keywords: EAs; evolutionary algorithms; GA; genetic algorithms; high integrity protection system; HIPS; metaheuristic optimisation; NLP; non-linear programming; non-deterministic polynomial time hard; NP-hard; optimisation test functions; PSO; particle swarm optimisation; SA; simulated annealing; safety systems.

Reference to this paper should be made as follows: Owa, K., Jackson, L. and Jackson, T. (201x) 'An intelligent novel tripartite - (PSO-GA-SA) optimisation strategy', *Int. J. Metaheuristics*, Vol. XX, No. YY, pp.ZZZ-ZZZ.

Biographical notes: Dr Kayode Owa received both BSc and MSc in the field of Computer Science and later obtained PhD from the University of Plymouth. He is currently a Research Associate with expertise in artificial intelligence, evolutionary and metaheuristics optimisation strategies at the Department of Aeronautical and Automotive Engineering, Loughborough University, UK.

Dr Lisa Jackson received PhD from Loughborough University and she is the Head of the Reliability Research Group and a Senior Lecturer in the Department of Aeronautical and Automotive Engineering, Loughborough University, UK.

Professor Tom Jackson received PhD from Loughborough University, he is a Professor of Information and Knowledge Management, Director of the Centre for Information Management and the Associate Dean [Research] at the School of Business and Economics, Loughborough University, UK.

This paper is a revised and expanded version of a paper entitled ‘An Improved Hybrid Evolutionary Optimization Approach’ presented by *the corresponding author at the 17th British-French-German Conference on optimisation from the 15th–17th June, 2015, Imperial college, London, United Kingdom.*

1 Introduction

The 21st century has witnessed an increased amount and the quality of research output in order to meet up with the numerous and diverse real-world challenges confronting us today. Many of these challenges can be classified as optimisation problems (Nesmachnow, 2014; Wang, Zhao and Ren, 2007; Rutenbar, 1989) and they exist in various fields and disciplines. A large number of these problems are prevalent in industries such as in control fields (Yan and Wang, 2015), clinical medicine and materials science (Lan et al., 2015), wind power energy generations (Rehman and Ali, 2015), manufacturing sectors (Liu et al., 2015), productions (Liu et al., 2014), processes (Singh et al., 2012), engineering sectors (Wu, 2012), and there are some other implementations which include missile evasion systems, trajectory planning and radar applications which can be found in the ministry of defence (MoD) (Lu, Miao and Li, 2013; Kumar and Singh, 2013). Pragmatically, such problems often involve making an almost impossible or impractical selection that gives a single global optimal solution which has the best fitness value to an optimisation problem. In a nutshell, the number of possible solutions (permutations/combinations) could run into millions or more depending on the problems; these are challenging tasks and non-trivial in nature; they are mostly highly non-linear, non-differentiable, non-convex, discontinuous and non-parametric systems.

As a result of this, there is a need to solve these complex and practical problems in order to arrive at the global optimal solution. Nesmachnow (2014) provided in his work an accurate overview of many cutting-edge approaches, elaborated on efficient available metaheuristic optimisation methods for solving difficult practical and theoretical problems and detailed the current and future research in the field. Methods and approaches such as the Quasi-Newton, Newton-Raphson, Gradient descent, Gauss-Newton, Levenberg-Marquardt and Nelder-Mead provide good solutions for solving convex optimisation problems especially when the problems have small dimensionality and possess only differentiable functions. However, there is the inability to effectively implement these approaches because of the higher probability of being trapped in a local minima solution.

The nature of these methods makes it unsuitable for solving non-differentiable functions and problems with high dimensions.

However, with the implementation usage of these robust and stochastic optimisation approaches, there are still some predominant issues such as the problem of the potential solution being trapped in a local minima solution space. Having mentioned the prevailing problems associated with the existing approaches, this research paper proposes a novel tripartite (PSO-GA-SA) optimisation approach. The main motivations for the research work are to propose a strategy that can conveniently search the solution space without getting stuck in a local optima solution. In addition, there is the necessity to efficiently solve and obtain quality optimum solutions as quickly as possible. The main contributions of this work are in the areas of parallel and strategic combinations of the three approaches: PSO, GA and SA; the algorithm is designed with the full potentials of both PSO and GA functioning simultaneously to exploit, explore and exchange good and relevant solution traits within the population and amidst themselves in real time with the solution of SA by improving on current solution in order to obtain better results without compromising the computational time. Moreover, there is an introduction of a high level of intelligent behaviour, whereby a flag is activated to avoid unnecessary function evaluations in the 'do while' loop of SA to reduce the computational time. Taking into account the various and numerous research already carried out in the area of metaheuristics optimisation, literature reveals that many approaches have been implemented using individual metaheuristics approaches; however, only a few works exist in the area of hybrid metaheuristics optimisation such as the PSO, GA and SA approaches. This proposed approach is designed to operate with high intelligence in order to explore diverse solution space, exploits good features of the individual methods in order to arrive at an optimum solution. The remaining part of this paper is organised as follows: Section 2 briefly reviews some existing evolutionary approaches, metaheuristic strategies and hybridisation processes. Section 3 explains the proposed PSO-GA-SA approach and its implementation. Section 4 discusses the experimental analysis and case study 1 (application to well-known mathematical test functions, its implementation, results and discussions). Section 5 details the case study 2; a real-world practical scenario, its descriptions and implementation with the obtained results while the conclusions is given in Section 6.

2 Metaheuristic strategies and hybridisation

2.1 Review of existing optimisation strategies

As mentioned in (Nesmachnow, 2014), some of these metaheuristic and evolutionary approaches include particle swarm optimisation (PSO), genetic algorithm (GA), simulated annealing (SA), ant colony optimisation (ACO), and artificial bee colony (ABC). There are advantages and disadvantages associated with each of these individual evolutionary/metaheuristic approaches. Many researchers have implemented all of these individual approaches; the extensive work of Chandrasekar and Ramana (Chandrasekar and Ramana, 2012) listed PSO, GA and SA as part of the best suited approaches to metaheuristics optimisation problems. Therefore, the historical details, advantages and disadvantages of PSO, GA and SA are discussed.

- *PSO*: PSO was proposed by Eberhart and Kennedy in 1995. Since then, numerous scientific research has been carried out and the technique has been applied to many

diverse problems (Eberhart and Kennedy, 1995; Aziz et al., 2011; Cagnina et al., 2008; Reynolds et al., 2005). PSO results do not improve considerably when the initial number of solution size is substantially large. PSO could be simple to design and converges rapidly to quality solutions with the potential of attaining a global optimal solution (Sharma and Singhal, 2015). PSO performs well when applied to optimisation problems of large dimensions and with non-differentiable functions.

- *GA*: GA was first developed by John Holland in 1960s. His original intention was not to use it for solving problems but to study about the principles of natural adaptation (Melanie, 1996). Research has found that the GA approach does not give good results when the initial number of solution size is too small. GA can conveniently explore for the global solution, but the implementation could be a challenge; the optimisation speed is rather slow and the convergence rate is very low (Sharma and Singhal, 2015).
- *SA*: SA was introduced by Kirkpatrick et al. in 1982 (Kirkpatrick, Gelatt and Vecchi, 1983; Rutenbar, 1989) for solving combinatorial optimisation problems. SA derives its utilisation from annealing in metallurgy where there is heating and manipulated cooling of a substances in order to diminish material defects and enhance the size of its crystals. SA is good for avoiding local minima solutions and it is very efficient for problems with a large search solution space.

2.2 *Hybridisation and related works*

The GA evolutionary approach (Melanie, 1996) is a very well established method in the literature. In addition, non-deterministic metaheuristic methods such as the PSO (Eberhart and Kennedy, 1995) and SA (Kirkpatrick, Gelatt and Vecchi, 1983; Rutenbar, 1989) are also very popular and have been used extensively to solve many problems. Since the numerous practical problems differ, this reality therefore leads to many approaches for solving many practical optimisation problems (Huesken, Jin and Sendhoff, 2002; Sbalzarini, Muller and Koumoutsakos, 2000). These approaches are either used individually or are combined with one another to provide better solutions to problems such as the work of (Hoseini and Shayesteh, 2010). However, the use of these approaches depends entirely on the specific problem application. There are also many recent approaches applied by researchers to improve the performances obtained from the use of evolutionary algorithms (EAs). For instance, the work of Zhao et al. (2014) adopted the use of an adaptive crossover and mutation to further improve the performance of obtaining better results in PSO-GA for solution layout problems.

In order to further make the outcomes and results of evolutionary and metaheuristic approaches more beneficial, combinations and hybrid versions of several methods are continuously being utilised. A recent work of Sharma and Singhal (2015) involved the insertion of the final optimal solution of the GA process into the PSO procedure. Also, the work of Yang et al. (2014) employed another strategy which involved the development of an improved version of a hybrid of PSO and SA by exploring the temperature cooling rate of SA in order to help in increasing the probability of obtaining a global optima solution. A hybrid version of the combinations of ACO, GA and SA (Hoseini and Shayesteh, 2010) was used for image contrast enhancement by the transformation of the input intensities.

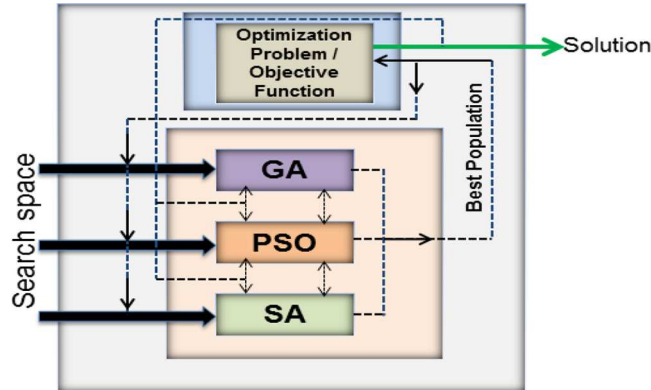
The work carried out by Chen and Chien in 2010 (Chen and Chien, 2010) explored the combination of ACO, GA, SA and PSO for solving the travelling salesman problem. Just as in the case of (Hoseini and Shayesteh, 2010), ACO was used to generate an initial solutions for GA-SA and then the result was further utilised by the PSO algorithm. The deficiencies of this approach are the lack of diversity using only the initial ACO solutions and the implementation of only one level of fitness evaluation in the entire process. In all these, no work is currently available that explores the combined simultaneous advantages of using PSO, GA and SA for optimisation; this approach also accommodates the inadequacies highlighted in (Hoseini and Shayesteh, 2010). This research paper designs a novel tripartite PSO-GA-SA approach that delivers improved results by simultaneously exploring the individual advantages of each of the approaches for more diversity; this further prevents the approach from being stuck in a local minimum solution.

3 Overview of the proposed PSO-GA-SA optimisation approach

3.1 Elements of the PSO-GA-SA tripartite method

This improved tripartite evolutionary optimisation approach is designed to simultaneously utilise the full advantages and good features of PSO, GA and SA approaches. Essentially, PSO can search for a diverse solution space at a fast rate; GA has the potential to arrive at the optimal solution but possibly at a slower rate while there is higher assurance of obtaining the global optimal solution with SA. The slow rate of temperature cooling in SA accounts for the additional computational time. The time involved in SA has made its application less appealing and this approach has taken this disadvantage into consideration. The tripartite operation in this proposed PSO-GA-SA approach has many advantages and features which are further highlighted. The initial and subsequent solutions are randomly generated in such a way to form solutions that span the whole range for a constrained optimisation problem; this is also more advantageous in the case of unconstrained optimisation. The approach is designed to use a minimal number of initial solutions so as to reduce the computational time. Good solutions are exchanged between and across PSO-GA-SA throughout the entire optimisation process. The traits exchanged are intelligent in such a way that the elite solution interacts across the board and evaluate fitness values and their suitabilities for solution; this creates room for more mating between each operation of PSO, GA and SA. The best solution attributes are transmitted and exchanged in such a way that each of the solution of PSO, GA and SA will present their best candidate for further propagation; this involves the exploitation and exploration of the populations for potential solutions. This process helps to avoid the operation being trapped in a local minimum solution. In this way, there is an efficient and effective use of the number of generations; this invariably leads to the drastic reduction in the number of generations for the proposed novel approach. Furthermore, there is an improved performance and a reduced computational time. This approach ensures a guaranteed global optimal solution. Figure 1 shows the pictorial view of the described proposed PSO-GA-SA approach; this is the process that occurs in a single generation where the best solution within the population is selected and individual solution fitness is evaluated through the objective function.

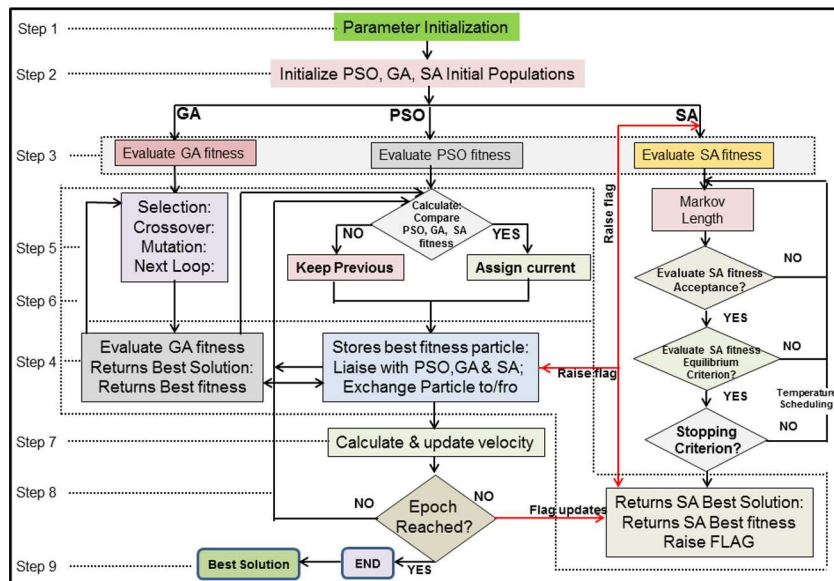
Figure 1 A three-way tripartite version combination of PSO, GA and SA (see online version for colours)



3.2 Methodology for the PSO-GA-SA approach

The steps, procedures and processes for the implementation of the novel PSO-GA-SA approach are enumerated. The flowchart for the novel PSO-GA-SA approach is shown in Figure 2 and are subsequently described in the following steps.

Figure 2 The flowchart for the novel PSO-GA-SA approach (see online version for colours)



Step 1 - Parameter initialisation: The process starts by deciding and choosing the following parameters: crossover, mutation, number of fitness evaluations (σ), number of generations (η), number potential member solutions in the population (ν) and minimum and maximum constraints on the population inputs. Other parameters defined and associated with the SA include Markov chain length, decay parameter, step factor, initial temperature and the tolerance value.

Step 2 - Population derivation: Initialise the swarm position population (a), swarm velocity population (b), swarm best position population (c) and genetic initial starting solution with all their populations confined within the upper and lower limits of their respective constrained values. The vectors (a), (b) and (c) depict the current position vector, attraction to the group best vector and attraction to the overall individual best, respectively. The SA is designed with a single initial random population; the algorithm is designed to further generate solutions in the neighbourhood of the initial starting population of solution as it commences to search for the problem solution.

Step 3 - Fitness evaluation: Obtain the fitness values of the potential solutions within the population generated in *step 2*.

Step 4 - Solution fitness exchange: The best n -dimension solution space vector/string in each of the populations in *step 3* is exchanged and transferred to/from one another; the best position particle (from PSO) joins together in the Tournament/Roulette selection process in GA before proceeding to the crossover/mutation mating; the best elite GA trait also replicates itself, combines and competes with the PSO particles; finally, SA also sends its best solution to both PSO and GA for fitness comparison and for further interactions while retaining the best in all the propagation scenarios. For increased efficiency, the initial random solution for the SA is replaced with the better solution from PSO and GA.

Step 5 - Diversity creation: All the processes in *step 4* are to ensure enough diversities throughout the entire process. New solutions are constantly being introduced in each generation from the PSO operations; and these are filtered across the GA and SA populations.

Step 6 - Intelligent behaviour: One of the admirable features of this algorithm is in the introduction of what is called an intelligent variable velocity. This is to further ensure an increased diversity and helps to avoid settling in a local optimal solution. The variable velocity (ρ) shown in Eq. (2) is introduced and injected into the algorithm in a situation where there has been no significant and successive improvement in the solution for a certain period of time. In addition, the algorithm carries out timing decisions with the help of the embedded intelligence throughout the entire process. For instance, the algorithm raises a Boolean flag whenever SA propagates through the temperature cooling procedure and produces its best candidate; this then enables the algorithm to know when it is appropriate to proceed along any prolong or unproductive path; this is made possible by retaining the knowledge of different performances of SA, GA and PSO. The SA algorithm is allowed to execute at least once in the entire process with the tolerance manipulations to enhance faster computations and time reduction; the algorithm compares the SA solution and introduces it into the PSO and GA population for further propagation depending on their individual fitness values. The value of the flag helps to determine the navigation through the algorithm for faster implementation.

Step 7 - Tripartite position update: The update fragment of the entire process of PSO, GA and SA is summarised in Eqs. (1)–(3).

$$\begin{aligned} Vel_i(k+1) = & \phi(k) * Vel_i(k) + \alpha_1 * \gamma_{1i} * [Pbest_i(k) - Pos_i(k)] \\ & + \alpha_2 * \gamma_{2i} * [Gbest(k) - Pos_i(k)] \end{aligned} \quad (1)$$

$$Vel_i(k+1) = \rho_\varphi * Vel_i(k+1) \quad (2)$$

$$Pos_i(k+1) = Pos_i(k) + Vel_i(k+1) \quad (3)$$

where:

i = particle index of the solutions within the population,

k = discrete time index,

$\alpha_{1_{\min}}$ is the minimum cognitive learning parameter,

$\alpha_{1_{\max}}$ is the maximum cognitive learning parameter,

$\alpha_{2_{\min}}$ is the minimum social learning parameter,

$\alpha_{2_{\max}}$ is the maximum social learning parameter,

Pos is the position vector,

Vel is the velocity vector,

$Pbest$ is the personal best vector,

$Gbest$ is the global best solution, and

γ_{1i} and γ_{2i} are the random numbers in the range between 0 and 1 interval that are, respectively, associated with the cognitive and social learning parameters.

Equation (1) is the tripartite velocity vector update equation, Eq. (2) is incorporated in order to create a varied diversity in each generation depending on the result performance while Eq. (3) is the tripartite position vector update equation. Equation (1) equates to the summation of the inertia, local memory (cognitive learning), and global memory (social learning). Equation (3) helps to constantly change and update the position vector in each generation as it gets closer to the optimal solution. In summary, the updates of Eqs. (1)–(3) are designed to contain all the solution traits existing and inherent in the PSO, GA and SA solution space.

The parameters in Eq. (1) are further explained in Eq. (4). The new velocity obtained in Eq. (1) is a linear combination of these three vectors (Pos , Vel and $Pbest$). The attractions to the group best and the global best help to drive the optimisation process to achieve an optimal solution.

$$\begin{aligned}\phi &= -(\phi_{\max} - \phi_{\min}) * \frac{\varphi}{\tau} + \phi_{\max} \\ \alpha_1 &= (\alpha_{1_{\max}} - \alpha_{1_{\min}}) * \frac{\varphi}{\tau} + \alpha_{1_{\min}} \\ \alpha_2 &= (\alpha_{2_{\max}} - \alpha_{2_{\min}}) * \frac{\varphi}{\tau} + \alpha_{2_{\min}}\end{aligned}\quad (4)$$

where:

the current iteration value = φ ,

the maximum iteration specified = τ ,

the inertia weight function = ϕ ,

the initial weight value for the inertia function = ϕ_{\min} ,

and the final weight value for the inertia function = ϕ_{\max} .

These initial values were determined heuristically after a few trials.

Step 8 - Next Generation: The procedure goes back and repeats from *step 3 up to step 7* by evaluating the fitness values of the new next generation populations.

Step 9 - Stopping Criterion: The algorithm stops when the specified number of epochs or generations is reached.

A pseudo-code describing the higher level schema of proposed PSO-GA-SA is provided.

Algorithm 1 Pseudocode for the Proposed Hybrid PSO-GA-SA

```

1: Initialize all parameters used by PSO, GA and SA
2: Initialize the starting number ( $\eta$ ) for the particles with random value position and velocity
   vector populations for PSO from input dimensions in the search space; this is the initial
   solution space or random initial populations
3: Initialize the starting number ( $\eta$ ) for the GA
4: Initialize a single random initial starting solution for the SA
5: Determine the Tolerance, Markov length, starting temperature, temperature decay factor
   value for the SA
   Evaluate the initial fitness values of all Initialized random ( $\eta$ ) GA possible solutions
6: GA:
7: Determine and select the next elite candidates
8: Perform crossover in order to select the next candidates
9: Perform mutation in order to select the next candidates
10: Evaluate the GA fitness
11: Determine the GA Best solution
12: Determine the GA Best fitness
13: Send Best fitness to PSO-GA-SA
14: Goto GA
   Evaluate the initial fitness values of all ( $\eta$ ) PSO possible solutions
15: PSO-GA-SA:
16: Receives and updates fitness values from GA, PSO and SA
17: Update the Best position vector with these candidates
18: Calculate the new velocity vector and update the new position vector using equations 1,
   2 and 3
19: Updates the Best solution
20: Keeps checking the Boolean SA flag status
21: Evaluate the PSO-GA-SA fitness
22: Determine the PSO-GA-SA Best solution
23: Determine the PSO-GA-SA Best fitness
24: Send Best PSO-GA-SA fitness back to PSO-GA-SA
25: Goto PSO-GA-SA
   Evaluate the fitness of the SA initial solutions
26: SA:
27: Sets the Boolean value of flag:
28: Receives and replaces the initial starting solution with a better solution from from GA
   and PSO.
29: while Absolute difference between current and new fitness > Tolerance do
30:   for Each Markov Length  $i$  do
31:     while (!
32:       if  $f(\vec{X}_i) \leq -f(\vec{P}_i)$  then
33:          $\vec{P}_i \leftarrow \vec{X}_i$ 
34:       if  $f(\vec{X}_i) < f(\vec{P}_g)$  then
35:          $\vec{P}_g \leftarrow \vec{X}_i$ 
36:       ) do
37:         Update the cooling Temperature with decay factor
38:         Update the Best solution
39:         Evaluate the SA fitness  $f(\vec{X}_i)$ 
40:         Update the SA fitness Evaluation
41:       Reset the Boolean value of flag:
42:       Send SA best fitness to PSO-GA-SA
43:     Goto PSO-GA-SA

```

4 Experimental analysis

4.1 Case study 1: optimisation test functions

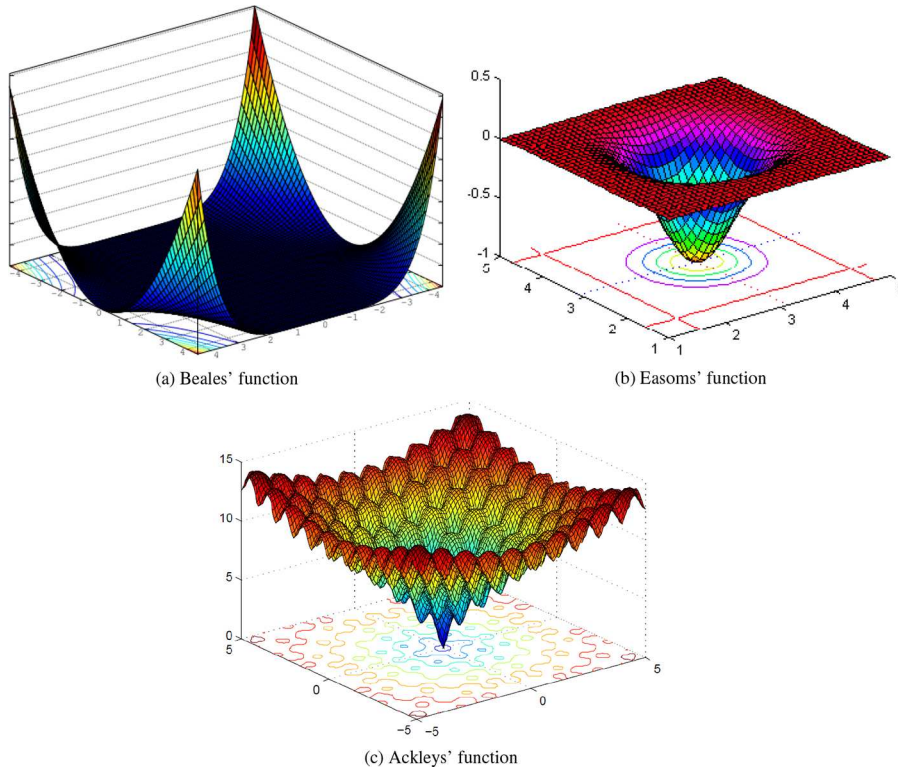
A total of five different approaches was designed in this work in order to test and compare the novel approach. They are (i) PSO, (ii) GA, (iii) SA, (iv) PSO-GA and (v) PSO-GA-SA. All the approaches were asserted with 3 standard benchmark optimisation test functions in the literature in order to ascertain the efficiency of this evolutionary global optimisation approach.

Equation (5) shown pictorially in Figure 3a, Eq. (6) illustrated in Figure 3b and Eq. (7) shown in Figure 3c are the Beales' function, Easoms' function and Ackleys' function, respectively.

$$f(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2 \quad (5)$$

$$\begin{aligned} \xi &= f(3.0, 0.5) = 0; \\ -4.5 &\leq x, y \leq 4.5 \end{aligned}$$

Figure 3 Optimisation test functions (see online version for colours)



$$f(x, y) = -\cos(x) \cos(y) \exp(-((x - \pi)^2 + (y - \pi)^2)) \quad (6)$$

$$\begin{aligned} \xi &= f(\pi, \pi) = -1; \\ -100 &\leq x, y \leq 100 \end{aligned}$$

$$f(x, y) = -20 \exp(-0.2\sqrt{0.5(x^2 + y^2)}) - \exp(0.5(\cos(2\pi x)) + (\cos(2\pi x)) + e + 20) \quad (7)$$

$$\begin{aligned} \xi &= f(0, 0) = 0; \\ -5 &\leq x, y \leq 5 \end{aligned}$$

4.2 Calibration analysis and parametrisation for case study 1

The choice of best parameterisation values is a very important issue and usually proves to be a complex task. Performing a systematic experiment with various parameter settings involves trying all different possible parameter values and their combinations which is also a difficult task. However, optimising parameter settings one by one does not lead to the optimal settings as most parameter tends to have effects on one another. The strategy adopted is to start from known and try different variations of parameters until an acceptable value is obtained. This section shows the results of the metaheuristics parameters used for the tripartite algorithm. Several trials were made in order to determine the best that will be taken forward to use. Calibration analysis and parametrisation tuning are performed. The parameters and variables investigated are as discussed. The value of Markov chain length was initially varied between 100 and 500. Results show that increasing the Markov chain length improves the results, but it considerably consumes a higher amount of time. Results also become much better by reducing the tolerance value from 1×10^{-2} to 1×10^{-6} but a low tolerance value makes the algorithm go through several fitness evaluations until the tolerance value is reached. Since too many fitness evaluations is not ideal for computation time, the best value which will give good result and that will not execute more than necessary fitness evaluation is chosen. For the PSO, the cognitive and social scaling parameters are very important; their choice determines whether the algorithm will be able to produce new solution that will move the position vector closer to the optimal solution. The number of fitness evaluations for algorithm involving SA varies depending on the starting random values. A typical value ranges between 10 and 300. The more we increase the starting initial solution space, generation number or the number of fitness evaluations, the better the results but it gets to a point where there are no further improvement in the results.

Default and standard values for crossover and mutation rate are used with little variations. The same parameters used for the individual approaches are also applied for the proposed tripartite approach.

The parameters and variables details are summarised as follows:

- 1 GA: Mutation rate = 0.2, crossover rate = 0.65, size of the initial starting solution space = 25, generation number = 25, number of fitness evaluations = 625

- 2 SA: Markov chain length = 100, decay parameter = 0.95, initial starting temperature = 100, tolerance value = 1×10^{-2} , size of the initial starting solution space = 1
- 3 PSO: size of the initial starting solution space = 13 each for both position and velocity random initialisation, generation number =25, number of fitness evaluations = 650, final weight for the inertia function =1.8, initial weight for the inertia function =1.6, minimum cognitive scaling parameter = 5, maximum cognitive scaling parameter = 4, minimum social scaling parameter = 1.5, maximum social scaling parameter = 4.
- 4 PSO-GA-SA: size of the initial starting solution space = 52, generation number =2, while the number of fitness evaluations is not predetermined before the commencement of the strategy.

4.3 Implementation, results and discussions

The aim of the optimisation algorithm is to minimise $f(x, y)$ and find the global(best) optimal solution from the search domains with specified criteria. Equations (5)–(7) are referred to as the objective or fitness functions. For Eq. (5): the search domain range for (x, y) is between -4.5 and 4.5 ; the local minima solution is located at the position coordinate $(3, 0.5)$ and the target minimum point (ξ) of $f(x, y) = 0$. For Eq. (6): the search domain range for (x, y) is between -100 and 100 ; the local minima solution is located at the position coordinate (π, π) and the target minimum point (ξ) of $f(x, y) = -1$ and also in a similar manner for Eq. (7), the search domain range for (x, y) is between -5 and 5 ; the local minima solution is located at the position coordinate $(0, 0)$ and the target minimum point (ξ) of $f(x, y) = 0$ (see Eqs. (5)–(7) for more details). Since each implementation of the EAs utilises different random solution space, the results are presented using the average values obtained over a period of 6 different scenarios. The results of the comparison of the various method results are given in Table 1 where μ is the average fitness deviations of all the three test functions, σ is the number of fitness evaluations, η is the number of generations used while ν is the number of initial starting solution used in order to obtain the optimal solution. The fitness deviations in Table 1 is calculated using the formula: $\{\mu = |f(x, y) - \xi|\}$ which is derived by obtaining the absolute value of the error difference between the fitness obtained ($f(x, y)$) and the target minimum point (ξ) . The summary of all the results when all the strategies were subjected to the same conditions (the values of σ and η) show the efficiency of the novel approach. The important features of this PSO-GA-SA approach are the efficient performance obtained with a reduced number of both fitness evaluations (σ) and generations (η). The algorithm is designed to produce optimal solutions effectively as from the second generation. Moreover, the algorithm takes into consideration the importance of time using intelligent flag mechanisms to decide and avoid unnecessary looping through the algorithm and as well producing an optimal solution without compromising the time expended.

Taking a closer look at the result of Ackleys' function (see Eq. (7) - Figure 3c). The two-dimensional Ackleys' function has a hole at the centre (global optimal solution) and the whole surface is covered with numerous local minima points and hence it poses a real practical challenge for optimisation algorithms. For instance, algorithms that use only gradient descent methods would be trapped in a local optima solution. This PSO-GA-SA approach is designed to search and evaluate a wider solution space in order to be able to manoeuvre across the various hills and valleys (local optima solutions); this algorithm combines the exploitative and exploratory components efficiently in order to obtain a better

result. The results show that the algorithm was able to arrive at a global optimal solution (See Table 1); the smaller the fitness deviation (μ) in the result table, the better the strategy. The PSO-GA-SA approach shows to have the smallest μ value and hence processes the best result.

Table 1 Performance table of the optimisation strategies using test functions

<i>Strategies performance</i>							
<i>Strategies</i>	<i>Beale</i>	<i>Easom</i>	<i>Ackley</i>	μ	ν	η	σ
GA	0.43	0.39	0.41	0.41	25	25	625
PSO	0.33	0.35	0.34	0.34	26	25	650
SA	0.31	0.33	0.29	0.31	1	1	598
PSO-GA	0.30	0.22	0.21	0.24	51	13	663
PSO-GA-SA	0.10	0.15	0.01	0.09	52	2	665

4.4 Statistical analysis

Statistical analysis is carried out in this section. In this paper, non-parametric test is preferred and used rather than the parametric tests; the reasons responsible for the preference are highlighted as follows:

We have a very small sample size of 6 data sample each while parametric test requires a sample size that is greater than 20. Another reason for using the non-parametric test is because there are outliers available in the data which you cannot afford to discard. Therefore, the Kruskal-Wallis test has been adopted in this work for the statistical analysis to determine if there are significant variations in the results of each of the optimisation approach. For the Kruskal-Wallis test, the sample size for each group should be at least 5. If a group has less than 5 observations, the p-value can be inaccurate. The p-value is the probability of finding the observed results when the null hypothesis (H_0) of a study question is true.

The Kruskal-Wallis test is a version of the independent test measures ‘One-Way’ analysis of variance (ANOVA) that can be performed on ordinal (ranked) data. In this work, we have 5 groups (GA, PSO, SA, PSO-GA and PSO-GA-SA) and 6 observations each for the result of the Beales function. Table 2 shows the Statistical analysis using the Kruskal-Wallis non-parametric test for the Beales test function whilst the ranking of the different 6 observations are provided in Table 3.

Table 2 Statistical analysis

<i>Observations</i>						
<i>Fitness deviations</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
GA	0.47	0.42	0.41	0.46	0.43	0.4
PSO	0.32	0.33	0.29	0.34	0.3	0.38
SA	0.3	0.27	0.28	0.28	0.37	0.35
PSO-GA	0.31	0.25	0.28	0.35	0.26	0.35
PSO-GA-SA	0.14	0.08	0.09	0.11	0.134	0.12

Table 3 Ranking

Ranking	Observations						T	T^2
	1	2	3	4	5	6		
GA	30	27	26	29	28	25	165	27225
PSO	17	18	13	19	14.5	24	105.5	11130.25
SA	14.5	9	11	11	23	21	89.5	8010.25
PSO-GA	16	7	11	21	8	21	84	7056
PSO-GA-SA	6	1	2	3	5	4	21	441

The task is to find if there is a difference between the 5 different optimisation strategies. The seven steps required in order to achieve, these include the following:

Step 1 - Define the null and the alternative hypotheses:

Null hypothesis (H_0); there is no difference between the different approaches.

Alternate hypothesis (H_1); there is a difference between the different approaches.

Step 2 - State the alpha value:

Using an alpha level of 0.001.

Step 3 - Calculate the degrees of freedom:

Degree of freedom (dof) = $k - 1$ where k is the number of approaches = 5. Therefore the dof is 4.

Step 4 - State the decision rules:

In stating the decision rules, we use the Chi-square table which contains the alpha level and the degree of freedom. The Chi-square table gives a value of 18.467. Therefore if the Chi-square value that will be calculated is greater than 9.48773, then we will reject the null hypothesis (there is a difference). If the calculation is less than the Chi-square value, then we will not reject the null hypothesis (there is no difference).

Step 5 - Calculate the test statistics:

Rank all the 30 observations from smallest to the highest. The the Chi-square calculation is given in Eq. (8).

Step 6 - State the results:

Using the Chi-square equation gives a value of 22.8333 which is slightly greater than 18.467.

Step 7 - State the conclusion:

The null hypothesis is therefore rejected; that means that there is a difference in the approaches and the proposed approach performed better than the rest of the individual methods.

$$H = \left(\frac{12}{N * (N + 1)} \right) * \left(\sum \frac{T_i^2}{n} \right) - 3 * (N + 1) \quad (8)$$

5 Case study 2: real-world problem scenario

The second case study is a real-world problem scenario that deals with the design of a safety protection system.

In order to properly demonstrate the capabilities of this novel approach for complexities and scalability, a safety system design known as the HIPS is utilised. This particular case study is considered a more practical optimisation design application for a real-life situation

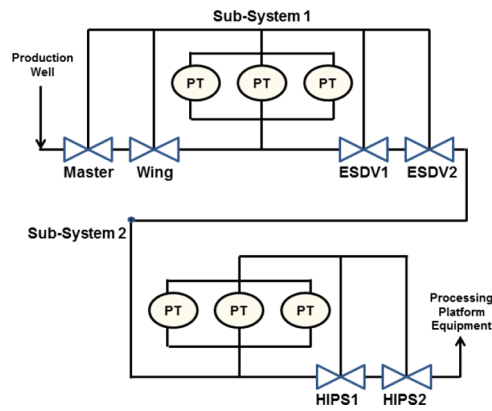
where there are a total number of 10 input variables in the design which render it more complex than the previous 2-variable case study. In addition, the novel approach is scalable to handle integer input apart from real decimal input as the initial starting solutions; the system is represented by a failure model which has a total number of almost 300 logic (AND, OR, Basic Events and House Events) gates in the design model which bring further complexities in the application and implementation approach.

5.1 HIPS parameter evaluation and implementation

5.1.1 HIPS design

The main function of the HIPS is to prevent a high-pressure surge passing through it. The HIPS is divided into two separate subsystems. Subsystem 1 ($SubS^1$) is the emergency shutdown (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 a Pressure Transmitter (PT). Subsystem 2 ($SubS^2$) provides an additional level of protection; the inclusion of the HIPS incorporates this second level of redundancy. In terms of their operation, the latter subsystem is completely independent of the first. Its method of protection is the same as the ESD system. Figure 4 illustrates this and represents the main features of the HIPS.

Figure 4 Structure of high integrity protection system (HIPS) (see online version for colours)



In this work, the design optimisation scheme has been applied to a HIPS where 10 variables are considered in the system design. These variables, their description and evaluation limits are shown in Table 4 where MTI is the maintenance or inspection test interval. Two points to note in the design:

- 1 It is assumed that whatever the valve type or the PT type are fitted, they are the same in the entire design.
- 2 The number of PTs required to activate the closure of valves on subsystem 1 or 2 is a function of the numbers installed (N_1, N_2) in the design (See Table 4).

Table 4 HIPS design variables

<i>Variables</i>	<i>Description</i>	<i>Range of values</i>
θ_1	MTI for $SubS^1$	1 week - 2 years
θ_2	MTI for $SubS^2$	1 week - 2 years
V	Valve type	1 or 2
P	PT Type	1 or 2
N_1	No of PTs fitted for $SubS^1$	1 - 4
N_2	No of PTs fitted for $SubS^2$	0 - 4
K_1	No of PTs required to activate $SubS^1$	1 - N_1
K_2	No of PTs required to activate $SubS^2$	0 - N_2
E	No of ESD valves fitted	0 - 2
H	No of HIPS valves fitted	0 - 2

Given the numbers of likely design variations that could result from all the 10 design variables described in Table 4, this amounts to almost 45 million possible safety system design options. It is therefore not feasible to completely solve for all the potential designs. Furthermore, it is an arduous task to understand the interactions that exist between all the design variables and is practically unimaginable for any engineer to manually design and test safety system for all these scenarios. The computerised optimisation strategy algorithm can easily step into play in order to determine the ‘best or optimal’ design option in an efficacious manner without going through the arduous task.

Each hardware component of the HIPS could either fail in a dormant mode or in a spurious manner. A dormant failure can be described as the inability of the component to carry out its desired task when occasion arises for it to operate as designed. In contrast, the spurious failure results from the component carrying out its desired normal function when its operation is not actually required; this can be termed as working inappropriately or functioning in a wrong working mode. Table 5 shows the details of the HIPS design variables: the failure rates (λ) and the mean time to repair - MTTR (ν) for each of the HIPS components in both the dormant and the spurious failure modes. These data will be used subsequently when calculating the unavailability and spurious trip probability of the HIPS. The hardware costs in £ for each component in the system as well as times taken to service each component at each maintenance test are also shown in Table 5. Each combination of HIPS variables gives a new system design with its particular features such as cost and the maintenance down-time (MDT). In this case, there are three limitations on the available resources:

- 1 The total cost of the system must be less than 1,000 units.
- 2 The average time each year that the system spends in the down state due to preventative maintenance is a maximum of 130 hours per year.
- 3 If the number of times that a spurious system shutdown occurs is more than once per year, then it is deemed unacceptable.

Table 5 Component failure data

Components	Dormant failures		Spurious failures		Cost (£)	Time (hours)
	λ (/hour)	ν (hours)	λ (/hour)	ν (hours)		
Wing valve	1.14×10^{-5}	36	1.00×10^{-6}	36	100	12
Master valve	1.14×10^{-5}	36	1.00×10^{-6}	36	100	12
HIPS valve 1	5.44×10^{-6}	36	5.00×10^{-7}	36	250	15
HIPS valve 2	1.00×10^{-5}	36	1.00×10^{-5}	36	200	10
ESDV valve 1	5.44×10^{-6}	36	5.00×10^{-7}	36	250	15
ESDV valve2	1.00×10^{-5}	36	1.00×10^{-5}	36	200	10
Solenoid valve	5.00×10^{-6}	36	5.00×10^{-7}	36	20	5
Relay contacts	0.23×10^{-6}	36	2.00×10^{-6}	36	1	2
PT 1	1.50×10^{-6}	36	1.50×10^{-5}	36	20	1
PT 2	7.00×10^{-6}	36	7.00×10^{-5}	36	10	2
Computer logic	1.00×10^{-5}	36	1.00×10^{-5}	36	20	1

5.1.2 HIPS cost evaluation

In calculating the total cost involved, explicit and implicit approaches are the two major classes of constraints required. Explicit ones can be determined and easily evaluated from an explicit function of the design variables. On the contrary, the 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 the following equations:

$$Cost = Cost(SubS^1) + Cost(SubS^2) \leq 1000 \quad (9)$$

$$Cost(SubS^1) = E(V_1C_{VE1} + V_2C_{VE2} + C_s) + N_1(P_1C_{P1} + P_2C_{P2}) + 261 \quad (10)$$

$$Cost(SubS^2) = H(V_1C_{VH1} + V_2C_{VH2} + C_s) + N_2(P_1C_{P1} + P_2C_{P2}) + 21 \quad (11)$$

where:

V_1 - valve type 1

V_2 - valve type 2

P_1 - pressure transmitter (PT) type 1

P_2 - pressure transmitter (PT) type 2

$C_{V1} = C_{VE1} = C_{VH1}$ - cost of the valve type 1,

$C_{V2} = C_{VE2} = C_{VH2}$ - cost of the valve type 2,

C_{P1} - cost of the pressure transmitter (PT) type 1,

C_{P2} - cost of the pressure transmitter (PT) type 2,

C_s - cost of the solenoid valves.

$E, H, N_1,$ and N_2 were described in Table 4.

The constant 261 (Eq. 10) for $SubS^1$ is the additional cost for the wing and master valve, their solenoid valves, the computer and control relays. $SubS^2$ has a fixed cost of 21 units Eq. (11) for the computer and control relays. From Eqs. (9)–(11), the cost of each system design is:

$$Cost = (E + H)(V_1C_{V1} + V_2C_{V2} + C_s) + (N_1 + N_2)(P_1C_{P1} + P_2C_{P2}) + 282 \quad (12)$$

5.1.3 HIPS maintenance down time evaluations

Similarly to the cost for the HIPS, average MDT (maintenance down time) is calculated as a sum of the maintenance down time of $SubS^1$ and $SubS^2$ for each of the potential designs:

$$MDT = MDT(SubS^1) + MDT(SubS^2) \leq 130 \quad (13)$$

$$MDT(SubS^1) = \frac{52}{\theta_1} [E(V_1M_{VE1} + V_2M_{VE2} + M_s) + N_1(P_1M_{P1} + P_2M_{P2}) + 47] \quad (14)$$

$$MDT(SubS^2) = \frac{52}{\theta_2} [H(V_1M_{VH1} + V_2M_{VH2} + M_s) + N_2(P_1M_{P1} + P_2M_{P2}) + 13] \quad (15)$$

where:

$M_{VE1} = M_{VH1} = M_{V1}$ - test time of the valve type 1,
 $M_{VE2} = M_{VH2} = M_{V2}$ - test time of the valve type 2,
 M_{P1} - test time of the pressure transmitter (PT) type 1,
 M_{P2} - test time of the pressure transmitter (PT) type 2,
 M_s - test time of the solenoid valve.

The expression $\frac{52}{\theta_x}$ in (Eqs. 14–15) gives the number of times the system is down in a year for the $SubS^1$ and $SubS^2$, respectively. The constant 47 (Eq. 14) is the sum of the test times for the wing and master valve, their solenoids, the computer and control relay for $SubS^1$. Similarly for $SubS^2$, the sum of the test time for the computer and control relay is 13 units (Eq. 15). From Eqs. (13–15), MDT for each design can be calculated using Eq. (16):

$$MDT = 52 * [H_1 * H_3 + H_2 * H_4 + H_5] \quad (16)$$

where:

$$H_1 = V_1M_{V1} + V_2M_{V2} + M_s,$$

$$H_2 = P_1M_{P1} + P_2M_{P2},$$

$$H_3 = \frac{E}{\theta_1} + \frac{H}{\theta_2},$$

$$H_4 = \frac{N_1}{\theta_1} + \frac{N_2}{\theta_2},$$

$$H_5 = \frac{47}{\theta_1} + \frac{13}{\theta_2}.$$

5.1.4 HIPS analysis

The most important feature of each safety system is its perfect operation and swift action when an urgent demand is required. Therefore, the objective is to minimise system unavailability, which means to minimise the probability of system failure on demand. Ideally, using the design alternatives, it is essential to determine which potential system design would produce the highest functionality and the minimal unavailability.

Practically, certain factors need to be taken into consideration and it is the available resources for this application. The limitations on resources were defined as cost, maintenance effort and the spurious frequency. The design options should improve the HIPS performance without violating the specified constraints. Consequently, the evaluation of each constraint is required in order to assess the overall desirability of each design option.

5.1.5 Fitness evaluation and penalty formulas

The system unavailability is the main optimisation criterion; However, resources are not inexhaustible. Therefore, the following limits were considered:

- $Cost \leq 1000$ units,
- Maintenance down time: $MDT \leq 130$ hours per year,
- Spurious system failure: $F_{sys} \leq 1$ per year.

If these three parameters exceed their respective limits, the following penalty equations were implemented in this PSO-GA-SA HIPS safety design:

- 1 The method utilised tries to form a direct relationship between cost and performance. *HIPS PSO-GA-SA* designs with excessive cost will not be adopted, so more the constraint violation the heavier the penalty. This is implemented using an exponential relationship of the form $y = x^{\frac{5}{4}}$. The cost penalty also takes into account both the cost violation and the system unavailability. This is achieved using a multiplying factor which, rather than being fixed, varies according to the system unavailability of the design. Hence, the modified cost penalty formula (C_p) for excess cost is shown in Eq. (17).

$$C_p = \left\{ \frac{(COST - 1000)}{1000} \right\}^{\frac{5}{4}} * Q_{sys} \quad (17)$$

where Q_{sys} is the system unavailability.

- 2 If the MDT of a particular *HIPS PSO-GA-SA* design exceeds 130 hours, a contribution is made to the unavailability of the system in the form of a penalty. The respective penalty (M_p) is shown in Eq. (18).

$$M_p = \frac{(MDT - 1000)}{8760} \quad (18)$$

where the value of 8760 is derived by multiplying 24 h by 365 days and this gives the number of hours per year.

- 3 The third constraint, excess spurious trip occurrence, is also related to the cost. If a spurious trip occurs, production ceases and hence a financial loss is incurred. It was assumed that the cost per hour for loss of production is 100 units. On average, a spurious trip requires 36 hours (refer to Table 5) to repair and only one such occurrence is acceptable in a yearly period. Using the cost penalty formula (Eq. 17), the spurious trip penalty (S_p) can be expressed as shown in Eq. (19):

$$S_p = \left\{ \frac{Excess\ Cost}{1000} \right\}^{\frac{5}{4}} * Q_{sys} \quad (19)$$

Each penalty is subsequently added to the system unavailability. The result is a sole fitness value for each design referred to as its penalised system unavailability (Q'_{sys}), shown in Eq. (20).

$$Q'_{sys} = Q_{sys} + C_p + M_p + S_p \quad (20)$$

5.2 Calibration analysis and parametrisation for case study 2

In a similar approach used for case study 1, the calibration analysis and parameterisation tuning for parameters and variables used for case study 2 are further discussed.

Since this is a more complex problem, just as in the previous case study similar explanations apply here as well. The value of Markov chain length was increased and initially varied between 350 and 1000. The tolerance value range from between 1×10^{-4} and 1×10^{-7} . A typical range value for SA is more difficult to determine in the more complex scenario but could be any where in the range between 10 and 3000. The number of initial starting solution space is not increased appropriately for the complex problem, since the optimisation design strategies seek to explore many generations in order to determine the efficiency and their ability not to get stuck in local minimal solutions.

The parameter and variable details are summarised as follows:

- 1 GA: Mutation rate = 0.2, crossover rate = 0.65, size of the initial starting solution space = 20, generation number = 500, number of fitness evaluations = 10000.
- 2 SA: Markov chain length = 1000, decay parameter = 0.95, initial starting temperature = 1000, tolerance value = 1×10^{-6} , size of the initial starting solution space = 1.
- 3 PSO: size of the initial starting solution space = 10 each for both position and velocity random initialisation, generation number = 500, number of fitness evaluations = 10000, final weight for the inertia function = 0.08, initial weight for the inertia function = 0.06, minimum cognitive scaling parameter = 0.02, maximum cognitive scaling parameter = 0.01, minimum social scaling parameter = 0.04, maximum social scaling parameter = 0.03.
- 4 PSO-GA-SA: size of the initial starting solution space = 41, generation number = 10, while the number of fitness evaluations is not predetermined before the commencement of the strategy.

The results of the number of fitness evaluations (σ) for all the strategies are provided in Table 6. The number of fitness evaluations for proposed PSO-GA-SA approach is the size of the initial starting solution space multiplied by the number of generations and added to the number of fitness evaluations in SA approach since this is executed only once.

Table 6 Characteristics and comparisons of the best design structure

<i>The subsystems</i>		<i>GA</i>	<i>PSO</i>	<i>SA</i>	<i>PSO-GA</i>	<i>PSO-GA-SA^a</i>
Sub Sys 1	No. of ESD valves (E)	1	1	1	1	1
	No. of PTs (N_1)	3	3	2	2	2
	No. of PTs to trip system (K_1)	1	2	2	1	1
	MTI (θ_1)	87	63	47	93	39
Sub Sys 2	No. of HIPS valves (H)	1	1	1	1	1
	No. of PTs (N_2)	3	2	3	2	1
	No. of PTs to trip system (K_2)	2	2	2	1	0
	MTI (θ_2)	110	75	34	86	47
Valve type (V)		1	1	1	1	1
PT type (P)		1	1	1	1	1
Maintenance down time (MDT) (hours)		128.53	127.90	129.53	129.11	129.62
Cost		644	632	622	599	575
Spurious trip (F_{sys})		0.4334	0.4382	0.4276	0.4076	0.15085
System unavailability (Q_{sys}) $\times 10^{-7}$		4.162	4.117	4.115	4.110	4.023
Number of fitness evaluations (σ)		10000	10000	8156	10000	8252

Source: ^a Proposed PSO-GA-SA approach

5.3 HIPS results and discussions

This section discusses the results of the implementation of the 5 optimisation strategies in the research for the optimisation of design of HIPS. However, this approach clearly further demonstrates the ability to use PSO-GA-SA for the HIPS design to find and explore for an optimal HIPS design to protect the equipment and the infrastructures. Table 6 gives the concise details of the result comparison for the 5 strategies. The included variable velocity in the algorithm also assisted not only to avoid being trapped in a local minima solution but to enhance obtaining a near optimal solution. The results discussion is further enumerated as follows:

- One of the aims of this approach is the effort introduced in order to reduce the number of generations since the PSO-GA-SA algorithm goes through a lengthier fitness evaluation as compared with the individual approaches: PSO, GA and SA. In one of the preliminary results, an interesting finding from the PSO approach is that it used a total of fitness evaluations ($\sigma = 4400$) where ($\eta = 220$) to produce the optimal result. In addition, it is even more exciting that this PSO-GA-SA approach rightly produced the exact same result with just fitness evaluations ($\sigma = 1710$) where ($\eta = 25$). This is as a result of the increased diversity in the proposed approach.
- Subsequent to the preliminary results and in order to further demonstrate the efficiency of this PSO-GA-SA approach, an improved results is later obtained in terms of the system unavailability (Q_{sys}) which is the actual parameter to be minimised. The reduced (Q_{sys}) value of 4.023×10^{-7} is better than the results obtained for all other approaches as shown in Table 6.

- In all of the optimisation approaches, only 1 quantity of valve of type 1 and pressure transmitter of type 1 were obtained for the PSO-GA-SA approach for subsystem 2. This is a cheaper design since it requires just one HIPS valve and only one pressure transmitter to be installed unlike the other approaches; a similar explanation can be articulated for the subsystem 1.
- Another key point to remember is the task to obtain the design that produces the best reduced overall cost. The cost implication of the design using PSO-GA-SA approach incurred a cost of £575 which is the cheapest cost obtained when compared with the rest of the results in Table 6.
- On the positive side, all the approaches produced similar results in terms of the maintenance down time (MDT) measured in hours. They are all within the range of between 127 hours and the maximum allowable of value of 130 hours.
- One of the main strength inferred from this approach is the ability to explore many diversified solution using best features of the three individual approaches.

These design comparison distinctions are clearly detailed in Table 6.

6 Conclusion

This paper presents the successful implementation of a novel tripartite optimisation approach. This novel approach uses the best features of the combination of PSO, GA and SA. PSO brings more varieties, diversity and faster convergence in obtaining the optimal solution; GA can search a wider space by exploring the mating capabilities of the probabilities of the crossover and mutation operators; and SA is known to guarantee the global optimal solution.

As can be seen the approach was first tested and validated using three 2-variable optimisation test functions: the Beales, Easoms and Ackleys' function; this proves the working principle of the approach. This GA-PSO-SA method was further extended, verified and applied to a more complex, real-life practical, 10-variable optimisation design of a HIPS. The HIPS is a safety system that can be installed in an industrial environment and on hazardous processes in order to mitigate occurrence such as overpressure, explosions, fires, explosions, releases of flammable or toxic materials, and vessel or a gas blowout. A fault tree analysis (FTA) was carried out for the HIPS design for offshore platforms in order to calculate the design unavailability to initiate a shutdown in the case of an emergency. The HIPS processes numerous design options and the optimisation tool is required in order to find the optimal system design within certain constraints.

The main metric for comparison in the entire strategies is the number of fitness evaluations. Altogether, the GA-PSO-SA approach was proved and tested to find better solutions with fewer fitness evaluations. Further testing also shows that the proposed approach does not stagnate or get stuck in a local solution after it is subjected to further fitness evaluations unlike the other approaches. This approach has been properly demonstrated to perform well with a reduced number of generations; it has also been conveniently shown to be suited for massive processes with large models such as HIPS that require faster optimisation and efficiency in materials, resource, power and energy utilisation. The design simplicity and the fast execution speed will likewise enhance its utilisation in varied industrial applications with simulations model designs.

Ultimately, the approach has the potential to be used further in more complex safety critical systems, where scalability is not deemed to be an issue. Further work will explore the multi-objective technique for the design of systems with more than one objective to be simultaneously satisfied and a non-dominated multi-objective pareto optimisation task in order to tackle prominent practical problems.

Acknowledgements

This work is supported financially by the Engineering and Physical Sciences Research Council (EPSRC) under the project titled ‘Adaptive Informatics for Intelligent Manufacturing (AI²M)’- (EPSRC Reference number is EP/K014137/1). This is a project carried out by the Department of Aeronautical and Automotive Engineering and the School of Business and Economics at the Loughborough University.

References

- Aziz, N., Mohemmed, A., Alias, M. and Aziz, K. (2011) ‘Particle swarm optimization for constrained and multiobjective problems: a brief review’, *Ipedr.Com*, Vol. 6, No. 1, pp.146–150.
- Cagnina, L.C., Esquivel, S.C. and Coello Coello, C.A. (2008) ‘Solving engineering optimization problems with the simple constrained particle swarm optimizer’, *Informatica*, Vol. 32, No. 3, pp.319–326.
- Chandrasekar, K. and Ramana, N.V. (2012) ‘Performance comparison of GA, DE, PSO and SA approaches in enhancement of total transfer capability using FACTS devices’, *Journal of Electrical Engineering and Technology*, Vol. 7, No. 4, pp.493–500.
- Chen, S-M. and Chien, C-Y. (2010) ‘A new method for solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques’, *Proceedings of the Ninth International Conference on Machine Learning and Cybernetics*, Qingdao, 11–14 July 2010, Vol. 5, pp.2477–2482. DOI: 10.1109/ICMLC.2010.5580809
- Eberhart, R. and Kennedy, J. (1995) ‘A new optimizer using particle swarm theory’, *MHS '95, Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Municipal Industrial Research Institute, Nagoya, Japan, October 4–6, pp.39–43. DOI: 10.1109/MHS.1995.494215
- Hoseini, P. and Shayesteh, M.G. (2010) ‘Hybrid ant colony optimization, genetic algorithm, and simulated annealing for image contrast enhancement’, *IEEE Congress on Evolutionary Computation (Cec)*, Vol. 5, No. 1, pp.4957–4960.
- Huesken, M., Jin, Y. and Sendhoff, B. (2002) ‘Structure optimization of neural networks for evolutionary design optimization’, *Proceedings of the Genetic and Evolutionary Computation Conference - Workshop*, New York, Vol. 9, No. 1, pp.13–16.
- Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983) ‘Optimization by simulated annealing’, *Journal of Science*, Vol. 220, No. 4598, pp.671–680.
- Kumar, P. and Singh, A.K. (2013) ‘Phase only pattern synthesis for antenna array using genetic algorithm for radar application’, *Indian Journal of Radio and Space Physics*, Vol. 42, No. 1, pp.259–264.
- Lan, W., Wen, Z., Hui-lin, Y. and Zong-ping, L.U.O. (2015) ‘Osteogenesis effect of biomaterials analyzed using topology optimization’, *Proceedings of the 34th Chinese Control Conference*, July 28-30, 2015, Hangzhou, China, Vol. 34, No. 1, pp.2765–2768.
- Liu, T., Gao, X. and Wang, L. (2014) ‘Study on multi-objective optimization of oil production process’, *Proceeding of the 11th World Congress on Intelligent Control and Automation Shenyang*, China, June 29–July 4, 2014, Vol. 7, No. 4, pp.1824–1829.

- Liu, H., Zhao, Q., Huang, N. and Zhao, X. (2015) 'Production line capacity planning concerning uncertain demands for a class of manufacturing systems with multiple products', *IEEE/CAA Journal of Automatica Sinica*, Vol. 2, No. 2, pp.217–225.
- Lu, Y., Miao, W. and Li, M. (2013) 'The air defense missile optimum target assignment based on the improved genetic', *Journal of Theoretical and Applied Information Technology*, Vol. 48, No. 2, pp.809–816.
- Melanie, M. (1996) *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Massachusetts, pp.5041–5046.
- Nesmachnow, S. (2014) 'An overview of metaheuristics: accurate and efficient methods for optimisation', *International Journal of Metaheuristics*, Vol. 3, No. 4, pp.320–347.
- Rehman, S. and Ali, S.S.A. (2015) 'Wind farm layout design using modified particle swarm optimization algorithm', *International Renewable Energy Congress*, Vol. 1, No. 6, pp.1–6.
- Reynolds, P.D., Duren, R.W., Trumbo, M.L. and Marks, R.J. (2005) 'FPGA implementation of particle swarm optimization for inversion of large neural networks', *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005, SIS 2005*, 2005, pp.389–392. DOI: 10.1109/SIS.2005.1501648
- Rutenbar, R.A. (1989) 'Simulated annealing algorithms: an overview', *IEEE Circuits and Devices Magazine*, Vol. 5, No. 1, pp.19–26.
- Sbalzarini, I.F., Muller, S. and Koumoutsakos, P. (2000) 'Multiobjective optimization using evolutionary algorithms', *Center for Turbulence Research. Proceedings of the Summer Program*, NASA Ames/Stanford University, pp.63–74.
- Singh, I., Qin, I., Xu, H., Huynh, C., Low, S., Clauberg, H., Chylak, B. and Acoff, V.L. (2012) 'Pd-coated Cu wire bonding technology: chip design, process optimization, production qualification and reliability test for high reliability semiconductor devices', *IEEE 62nd Electronic Components and Technology Conference*, Sheraton San Diego Hotel and Marina San Diego, CA, USA, 29 May–1 June 2012, pp.1089–1096.
- Sharma, J. and Singhal, R.S. (2015) 'Comparative research on genetic algorithm, particle swarm optimization and hybrid GA-PSO', *International Conference on Computing for Sustainable Global Development (INDIACom)*, Vol. 2, No. 1, pp.110–114.
- Wang, Y-X., Zhao, Z-D., Ren, R. (2007) 'Hybrid particle swarm optimizer with tabu strategy for global numerical optimization', *IEEE Congress on Evolutionary Computation*, Vol. 2, No. 1, pp.2310–2316.
- Wu, J-Y. (2012) 'Stochastic global optimization method for solving constrained engineering design optimization problems', *Sixth International Conference on Genetic and Evolutionary Computing*, Kitakyushu, Japan, 25 Aug–28 Aug 2012, pp.404–408.
- Yan, Z. and Wang, J. (2015) 'Nonlinear model predictive control based on collective neurodynamic optimization', *IEEE Transaction on Neural Networks and Learning Systems*, Vol. 26, No. 4, pp.840–850.
- Yang, H., Yang, Y., Yang, Z. and Zhang, L. (2014) 'An improved particle swarm optimization algorithm based on simulated annealing', *International Conference on Natural Computation*, Vol. 10, No. 1, pp.529–533.
- Zhao, F., Li, G., Hu, H., Du, J., Guo, C. and Li, T. (2014) 'A novel improved hybrid particle swarm optimisation based genetic algorithm for the solution to layout problems', *Proceedings of the 11th World Congress on Intelligent Control and Automation*, Shenyang, China, 29 June–4 July 2014, Vol. 11, No. 1, pp.5041–5046.