

Multiple-Objective Optimization and Design of Series-Parallel Systems Using Novel Hybrid Genetic Algorithm Meta-Heuristic Approach

Essa Abraham Abdulgader Saleem*, Thien-My Dao, Zhaoheng Liu

Department of Mechanical/Manufacturing Engineering, École de Technologie Supérieure, Montreal, Canada

Email: *essa-abraham-abdulgader.saleem.1@ens.etsmtl.ca, *essasaleem25@hotmail.com

How to cite this paper: Saleem, E.A.A., Dao, T.-M. and Liu, Z.H. (2018) Multiple-Objective Optimization and Design of Series-Parallel Systems Using Novel Hybrid Genetic Algorithm Meta-Heuristic Approach. *World Journal of Engineering and Technology*, 6, 532-555.

<https://doi.org/10.4236/wjet.2018.63032>

Received: April 26, 2018

Accepted: June 9, 2018

Published: June 12, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this study, we develop a new meta-heuristic-based approach to solve a multi-objective optimization problem, namely the reliability-redundancy allocation problem (RRAP). Further, we develop a new simulation process to generate practical tools for designing reliable series-parallel systems. Because the RRAP is an NP-hard problem, conventional techniques or heuristics cannot be used to find the optimal solution. We propose a genetic algorithm (GA)-based hybrid meta-heuristic algorithm, namely the hybrid genetic algorithm (HGA), to find the optimal solution. A simulation process based on the HGA is developed to obtain different alternative solutions that are required to generate application tools for optimal design of reliable series-parallel systems. Finally, a practical case study regarding security control of a gas turbine in the over-speed state is presented to validate the proposed algorithm.

Keywords

Multi-Objective Optimization, Reliability-Redundancy Allocation, Overspeed, Gas Turbine, Hybrid Genetic Algorithm

1. Introduction

Optimization of series-parallel systems is an important aspect of equipment design strategies. The optimized system characteristics, such as reliability, cost, weight, and volume contribute toward designing the best machine. This approach is challenging because the reliability needs to be maximized whereas the other objective functions need to be minimized. In practice, system reliability optimization is critical, and over the last two decades, considerable effort has been devoted toward the development of reliability criteria for quantifying the

nature of generation, transmission, and circulation in composite system frameworks. To improve component reliability and implement redundancy while achieving a trade-off between system performance and resources, reliability design that aims to establish an optimal system-level configuration has long been considered an important advantage in reliability engineering. At present, system reliability is of considerable research significance, as engineering fields involve continual advancements in fixed systems and applications with increasing levels of complexity. Thus, it is imperative for production systems to perform satisfactorily during their expected lifespan. However, failure is an inevitable phenomenon associated with technological advancement of the equipment used in various industries. The reliability-redundancy allocation problem (RRAP) has been studied to optimize system reliability on the basis of the redundancy allocation problem (RAP) [1]. The RRAP has attracted considerable attention from the viewpoint of developing heuristic optimization algorithms. This paper focuses on an RRAP with the objective of maximizing system reliability under nonlinear constraints, such as system cost, weight, and volume. The RRAP has been shown to be an NP-hard problem, and various optimization approaches have been proposed to solve it. These methods, which are called meta-heuristic methods, have been widely researched and implemented. They can obtain feasible solutions within limited computing time. The main goal of RRAPs is to select the levels of redundancy and component reliability for maximizing and improving system reliability and performance. RRAPs are useful for designing not only systems that are taken together on a large scale but also systems produced in large-scale industrial operation using off-the-shelf components.

2. Literature Review

A reliability-redundancy optimization problem can be formulated using components and levels of redundancy to maximize some objective function, given system-level constraints on reliability, cost, and/or weight. The problem of maximizing system reliability through redundancy and component reliability selection is called the reliability-redundancy allocation problem (RRAP). Reliability optimization has been the subject of several studies by Kuo *et al.* [1], [2], [3]. Forsthoffer [4], Kundur [5], Hejzlar [6], and Seebgreets [7] conducted studies on overspeed protection, such as analysis of the instability of steam turbines and analysis of the reliability of wind turbines. Dhingra [8] developed an application of the reliability-redundancy optimization problem with regard to overspeed protection by using a multi-objective approach to maximize system reliability and minimize consumption of resources (cost, weight, and volume). This approach involves a goal programming formulation and a goal achievement method for generating Pareto optimal solutions. Control and overspeed protection for a gas turbine are nearly the same as those for a steam turbine. A gas turbine operates at a higher temperature than a steam turbine; hence, it requires closer control, called control sequencing. Sequencing allows automatic control of the gas tur-

bine. Fetanat *et al.* [9] proposed an optimal design for control and overspeed protection of gas turbine by means of reliability-redundancy optimization achieved using a new type of harmony search algorithm (HSA) known as the elitism Box-Muller harmony search algorithm (EBMHSA). Dhingra and Rao [10] used goal programming and goal attainment formulations under fuzziness in a multi-objective reliability apportionment problem subject to several design constraints. Rao proposed three methods for finding the optimal solution of each objective function: a method for determining reliability, a method for minimizing cost, and a method for controlling weight. Rao's approach has been developed to optimize redundant series-parallel systems, all components of which are time-dependent. The proposed model for simulation is the overspeed control system for a gas turbine engine. This model, proposed by Dhingra, is a combination of mechanical and electrical systems. Overspeed control is the first step against excessive speed. In general, the emergency reset of the system is designed independent of the overspeed control. Hence, high-reliability operation of control valves is considered. In the normal working mode, the control valves are opened sequentially [5]. Luus [11] proposed a new non-linear integer programming method that considers the component reliability to be fixed. However, a more general problem is one where the optimal redundancy in each stage is determined to obtain the maximum system reliability. To solve the RRAP, several global optimization methods as well as heuristic and meta-heuristic methods have been proposed in the literature, including the Lagrangian multiplier method, branch and bound method, and linear programming [3] [8] [12] [13]. These approaches do not guarantee exact optimal solutions but achieve reasonably good solutions for complex problems with relatively short computing time. Heuristic techniques, including genetic algorithms, require derivatives for all non-linear constraint functions, which are not derived easily because of the high computational complexity. Yokota *et al.* [14] and Hsieh *et al.* [15] applied genetic algorithms (GA) to mixed-integer reliability optimization problems. Zhao *et al.* [16] developed a hybrid GA with a flexible allowance technique for solving constrained engineering design optimization problems. Kanagaraj *et al.* [17] and Ghodrati and Lofti [18] developed a hybrid cuckoo search (CS)/GA algorithm to solve reliability-redundancy optimization problems and global optimization problems, respectively. Gen and Yun [19] developed a soft computing approach for solving various reliability optimization problems. This method combines rough search techniques and local search techniques to prevent premature convergence of the solution. Zou *et al.* [20] proposed a global harmony search algorithm for solving bridge and overspeed protection system optimization problems by combining the harmony search algorithm with concepts from particle swarm optimization. Different programming and evolutionary optimization techniques have been adopted to optimize different types of RRAPs, e.g., GA [15] and a new interpretation and formulation of the RRAP [21] using a new mixed strategy and a modified version of the genetic algorithm (MVGA), which shows distinct ad-

vantages compared to traditional approaches. Afonso *et al.* [22] proposed a modified version of the imperialist competitive algorithm (ICA) and demonstrated its capabilities by comparing its results with the best-known results of different benchmarks. Quy [23] developed a new method to optimize a multi-objective model in certain mechanical systems by using the fuzzy multi-objective method. His approach is based on the algorithm proposed by Rao [10], and he applied it to the modeling and analysis of the overspeed control system of a gas turbine engine. All the components of this model are time-dependent. The performance of the algorithm was verified by programmed simulation of the above-mentioned model. In summary, Dhingra [8], Rao and Dhingra [10], and Quy [23] developed effective multi-objective fuzzy optimization techniques for engineering design. In particular, they adopted fuzzy programming, which is a powerful technique for solving optimization problems with fuzzy parameters. However, the use of uncertain information for reliability allocation requires further investigation. Moreover, they treated component risk/cost functions as continuous. Thus, no general method for solving the component reinforcement problem with discontinuous risk/cost functions has been proposed thus far. In addition, the three above-mentioned studies did not adopt any random-search-based global optimization methods. In other words, the entire family of meta-heuristics that can efficiently solve highly nonlinear nonconvex mixed-integer optimization problems has been overlooked. The major drawback of these studies is that none of them has developed a practical tool for designing actual components with distinct physical properties, such as cost, weight, volume, and reliability. In this paper, we present a hybrid GA (HGA) approach based on the redundancy allocation problem to find the number of redundant components that either maximize reliability or minimize cost, weight, and volume under various resource constraints. The computational results of our approach are compared with those of previously proposed algorithms.

3. Reliability-Redundancy Allocation Problems (RRAPs)

In this study, a reliability-redundancy allocation problem of minimizing the multi-objective function $[-f_1, f_2, f_3]$ subject to several nonlinear design constraints can be stated as a nonlinear mixed-integer programming model. The multi-objective formulation was obtained by applying cost and weight constraints to an objective function. In other words, the general problem of reliability and redundancy is assigned to each of the subsystems such that the system reliability, cost, and weight are optimized. The problem is overspeed protection of a gas turbine system with a time-related cost function, and the multi-objective RRAP model is as follows:

$$\max Rs(r, n) \& \min Cs(r, n) \& \min Ws(r, n)$$

$$\text{Subject to : } g_j(r, n) \leq a_j, j = 1, \dots, m$$

$$1 \leq n_i \leq 10, i = 1, 2, \dots, 4, n_i \in Z^+$$

$$0.5 \leq r_i \leq 1 - 10^{-6}, r_i \in r^+$$

Many designers have attempted to improve the reliability of manufacturing systems or product components for greater competitiveness in the market. Typical approaches for achieving higher system reliability include increasing the reliability of system components and using redundant components in various subsystems of the system [2] [15].

4. Mathematical Formulation of the Problem

The mathematical model of the optimization problem is given by the equations below. The system reliability, cost, weight, and product of weight and volume are constrained by the design. The resulting multi-objective reliability apportionment problem is as follows: find n and r that minimize the multi-objective function $[-f_1, f_2, f_3]$ subject to $g_j(r, n) \leq a_j, j = 1, \dots, m$. **Figure 1** shows a typical example of a series-parallel system configuration with k-out-of-n subsystem reliabilities where

$$f_1(r, n) \text{ is the system reliability} \tag{1}$$

$$f_2(r, n) \text{ is the total system cost} \tag{2}$$

$$f_3(r, n) \text{ is the total system weight} \tag{3}$$

Subject to

$$g_1(r, n) = \sum_{i=1}^4 v_i n_i^2 \leq 250 \text{ (Volume)} \tag{4}$$

$$g_2(r, n) = \sum_{i=1}^4 w_i n_i \exp\left(\frac{n_i}{4}\right) \leq 500 \text{ (Weight)} \tag{5}$$

$$g_3(r, n) = \prod_{i=1}^4 [1 - (1 - r_i)^{n_i}] \geq 0.95 \text{ (Reliability)} \tag{6}$$

$$g_4(r, n) = \sum_{i=1}^4 \alpha_i \left(\frac{-t}{\ln r_i}\right)^{\beta_i} \left[n_i + \exp\left(\frac{n_i}{4}\right)\right] \leq 400 \text{ (Cost)} \tag{7}$$

$$1 \leq n_i \leq 10, i = 1, 2, 3, 4, [g_5(r, n)], n_i \in Z^+ \tag{8}$$

$$0.5 \leq r_i \leq 0.999999, [g_6(r, n)], r_i \in r^+ \tag{9}$$

where

$$f_1 = \max R_s(r, n) = \prod_{i=1}^4 [1 - (1 - r_i)^{n_i}] \tag{10}$$

$$f_2 = \min C_s(r, n) = \sum_{i=1}^4 \alpha_i \left(\frac{-t}{\ln r_i}\right)^{\beta_i} \left[n_i + \exp\left(\frac{n_i}{4}\right)\right] \tag{11}$$

$$f_3 = \min W_s(r, n) = \sum_{i=1}^4 w_i n_i \exp\left(\frac{n_i}{4}\right) \tag{12}$$

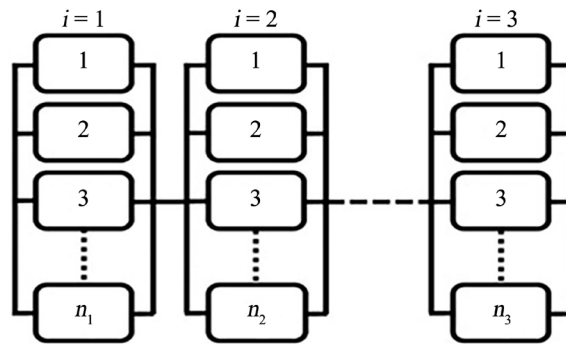


Figure 1. General series-parallel redundancy system.

Notation

r_i : Reliability of component in subsystem i ;

n_i : Number of redundant components in subsystem i ;

r, n : Vectors of r_i and n_i ;

R_s : System reliability;

N : Number of subsystems in the system;

f_1 : Objective function for system reliability;

f_2 : Objective function for system cost;

f_3 : Objective function for system weight;

$g_j(\cdot)$: Constraint function # j ;

a_j : Constraint limit # j ;

m : Number of constraints.

5. Methodology Framework

This study aims to propose a new algorithm that can be applied to optimization problems such that system reliability is maximized while system cost and system weight are minimized. The reliability, cost, and weight are subject to four nonlinear resource constraints, and the optimal levels of the reliability of component, r_p , and the number of redundant components, n_p , are to be determined at each stage i of the mechanical system.

Before introducing the RRAP, we present the following assumptions and notations that have been used throughout the entire paper. The hybrid function allows the optimization algorithm to identify the solution of the redundancy problem that achieves the optimal trade-off between the optimization objectives from several optimal solutions. We performed 10 simulations for every experiment and used the best result among the 10 reliability values obtained. The best configuration of each point corresponding to the largest reliability value is given with the corresponding cost, weight, and weight values.

Assumptions

- The supply of components is unlimited.
- The weight and volume of the components are known and deterministic.
- All the redundant components of individual subsystems have different values, and every branch of the system has a different number of components.

- The failure rate of the components in each subsystem is constant.
- Failed components do not damage the system and are not repaired.
- All redundancies are active: the hazard function is the same regardless of whether it is in use.
- Failures of individual components are independent of one another but dependent on the number of working elements.

As mentioned above, few studies have reported the use of HGA for reliability allocation optimization with time-dependent reliability. We need to check whether our approach of using only HGA can guarantee the location of the optimal solution and whether the final solution obtained by the proposed HGA is superior to that obtained by existing methods.

Figure 2 shows the flowchart of the proposed algorithm. The HGA procedures that implement our methodology are illustrated. The proposed algorithm involves the following steps:

- 1) Define the functions of the design problem (R_s , W_s , V_s , and C_s).
- 2) Define the nonlinear constraints.
- 3) Define the lower bound and upper bound for r_i and n_i .
- 4) Chose the optimization algorithm (fmincon, fminmax, GA, and HGA).
- 5) Solve the optimization problem.
- 6) Calculate the optimal values (R_s , C_s , and W_s).

The hybrid GA is a combination of fmincon and GA. GA is used to find the global optima for optimization problems. “Fmincon” uses gradient information to facilitate rapid convergence. “HybridFcn” allows the GA to find the valley containing the global minimum. Then, fmincon is used to rapidly obtain the minimum of this valley. A hybrid function is an optimization function that runs after the GA terminates in order to improve the value of the fitness function. The hybrid function uses the final point from the GA as its initial point.

This study consists of two parts. In the first part, we identify the approach for solving the problem by using MATLAB code and compare the results with previous results [23] to determine the number of redundant components in stage i and the reliability for each component. The second part involves a novel contribution: we develop a model for the entire system with the desired level of reliability. Specifically, we develop a simulation procedure and implement it with different numbers of components for each stage with different values of each component. We use this novel approach to determine the converged value of system reliability until we obtain the values of n_i and r_i corresponding to value of the maximum reliability. Toward this end, we need to perform optimization. For the general structure of the network, we fixed the system reliability to a certain level, *i.e.*, greater than or equal to 0.95.

We implemented a single-objective function with nonlinear constraints and tested it using two methods (n_i is an integer in our problem). The results are summarized in **Table 1** and **Table 2**. In addition, we implemented a multi-objective function. The initial results were obtained for four functions, and r_i and n_i were randomly set to evaluate each function.

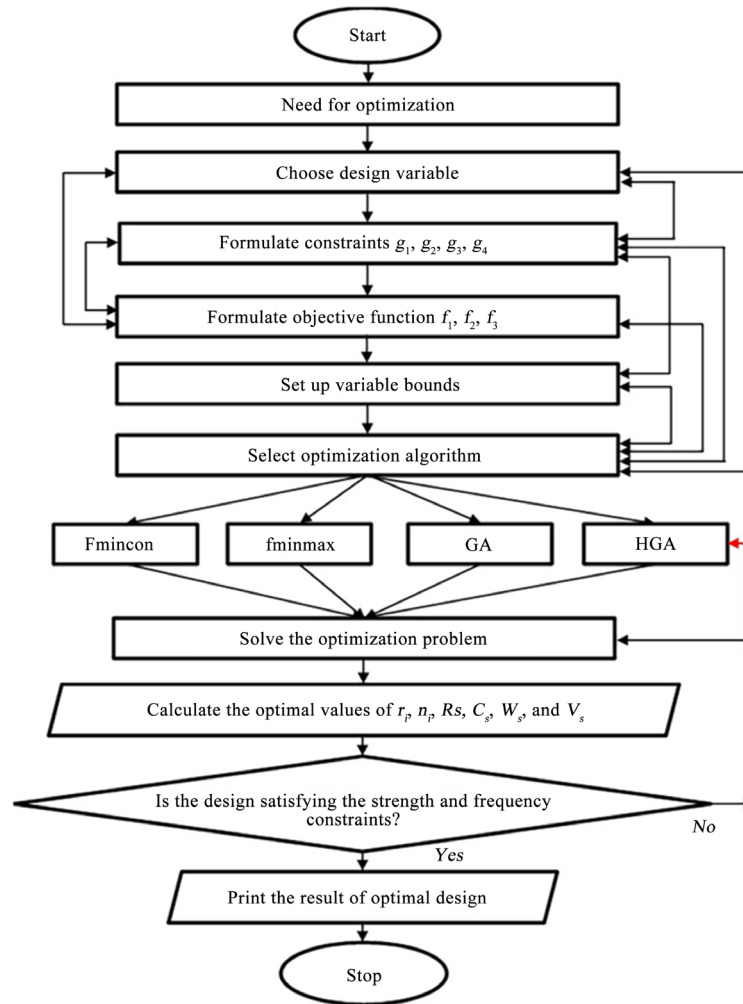


Figure 2. Flowchart of proposed simulation procedure.

Table 1. Simulation results for single-objective function using fmincon optimization method.

Objective	Stage	Reliability	Component	Simulation Result
Maximize System Reliability	1	0.8998	5	Rs = 0.9999
	2	0.8680	6	Cs = 419.2534
	3	0.9439	4	Ws = 541.2671
	4	0.8728	6	Vs = 217
Minimize System Cost	1	0.5846	5	Rs = 0.9439
	2	0.5184	6	Cs = 36.0616
	3	0.6988	4	Ws = 475.1981
	4	0.5252	5	Vs = 195
Minimize System Weight	1	0.9534	1	Rs = 0.9232
	2	0.9313	2	Cs = 422.7688
	3	0.9770	1	Ws = 60.8431
	4	0.9351	2	Vs = 20
Multi-objective Functions	1	0.8493	3	Rs = 0.9740
	2	0.7980	3	Cs = 109.3931
	3	0.9147	2	Ws = 147.0485
	4	0.8060	3	Vs = 57

Table 2. Simulation results for single-objective function using fminimax optimization method.

Objective	Stage	Reliability	Component	Simulation Result
Maximize System Reliability	1	0.9001	5	Rs = 0.9999
	2	0.8685	6	Cs = 420.2802
	3	0.9431	4	Ws = 541.2671
	4	0.8732	6	Vs = 217
Minimize System Cost	1	0.5846	5	Rs = 0.9439
	2	0.5184	6	Cs = 36.0616
	3	0.6988	4	Ws = 475.1981
	4	0.5252	5	Vs = 195
Minimize System Weight	1	0.9534	1	Rs = 0.9232
	2	0.9313	2	Cs = 422.7688
	3	0.9770	1	Ws = 60.8431
	4	0.9351	2	Vs = 20
Multi-objective Functions	1	0.8493	3	Rs = 0.9740
	2	0.7980	3	Cs = 109.3850
	3	0.9148	2	Ws = 147.0485
	4	0.8059	3	Vs = 57

First step: We implemented a multi-objective function, and we defined the general objective function as follows:

$$f = 10f_1 + f_2/400 + f_3/500 ; \text{ (new definition)}$$

The above-mentioned has three parts: reliability, cost, and weight. This equation maximizes reliability but minimizes cost and weight. It is a normalized form of the objective function because we consider the upper bound of each objective. We penalized the reliability (with a value of 10) for greater emphasis. In addition, we set the upper bounds for Cs and Ws as 400 and 500, respectively. Therefore, if we divide by these values and take the sum, we will always get a number less than one. Thus, we normalized the functions (f_1 , f_2 , and f_3).

Second step: We used fmincon and fminmax to solve the objective function.

Third step: We used the GA toolbox and applied this algorithm to our single- and multi-objective function problems. The results are summarized in **Table 3**.

Fourth step: We applied the GA to a new type of multi-objective function and evaluated the results.

Fifth step: We applied the global multi-objective GA to the problem and obtained 70 sets of Pareto optimal solutions.

Last step: We applied HGA optimization to single- and multiple-objective functions on the basis of our first approach. The results are summarized in **Table 4**.

Our multi-objective function aims to minimize cost and weight in the first approach. The results of our optimization give us n_i and r_i for each stage as well as for the entire system, as shown in the final result table. In this study, we performed optimization using GA and HGA. We used the same approach as that for obtaining a constrained minimum of a scalar function of several variables starting at an initial estimate. This is generally referred to as constrained nonlinear optimization or nonlinear programming (fmincon). We used different op-

timization approaches and finally used HGA. Specifically, we employed GA and fmincon to implement HGA using the first approach by varying r_i and n_i to achieve the desired system reliability with the objective function. Further, we fixed the system reliability R_s to obtain a system with minimum cost and weight in order to determine the structure of our new design in the second approach, which minimizes the worst-case value of a set of multivariable functions, starting at an initial estimate. The values may be subject to constraints. This is generally referred to as the *minimax* problem (fminmax).

We also varied the level of system reliability to show how we can select the desired system reliability; accordingly, we can change the structure of the entire system. In this step, we used GA and MATLAB toolbox. Here, we do not maximize the system reliability R_s but we want $R_s = A$, and we want to determine the system structure for achieving the minimum cost and weight. We assumed that n_i is a continuous value. In this case, the first method of optimization using fmincon is summarized in **Table 5**. In addition, we can see the result of the second approach of optimization, *i.e.*, fminmax. The results of our contribution are summarized in **Tables 6-8**, which show the different values obtained after we fixed the system reliability.

We tested various algorithms to identify the best ones, which were found to be GA or HGA.

Table 3. Simulation results for single-objective function using GA optimization method.

Objective	Stage	Reliability	Component	Simulation Result
Maximize System Reliability	1	0.8902	5	$R_s = 0.9999$
	2	0.8603	6	$C_s = 389.3556$
	3	0.9500	4	$W_s = 475.1981$
	4	0.8806	5	$V_s = 195$
Minimize System Cost	1	0.6106	5	$R_s = 0.9501$
	2	0.5550	5	$C_s = 37.4312$
	3	0.6509	5	$W_s = 471.1963$
	4	0.5465	5	$V_s = 200$
Minimize System Weight	1	0.8977	2	$R_s = 0.9511$
	2	0.9537	2	$C_s = 414.0766$
	3	0.9732	1	$W_s = 72.9236$
	4	0.8987	2	$V_s = 23$
Multi-objective Functions	1	0.8504	3	$R_s = 0.9740$
	2	0.7956	3	$C_s = 109.1462$
	3	0.9167	2	$W_s = 147.0485$
	4	0.8049	3	$V_s = 57$

Table 4. Simulation results for single-objective function using hybrid optimization method.

Objective	Stage	Reliability	Component	Simulation Result
Maximize System Reliability	1	0.8971	5	$R_s = 0.9999$
	2	0.8659	6	$C_s = 381.5582$
	3	0.9358	4	$W_s = 475.1981$
	4	0.8769	5	$V_s = 195$

Continued

Minimize System Cost	1	0.7997	4	Rs = 0.9939
	2	0.7896	4	Cs = 133.4582
	3	0.7154	5	Ws = 346.2031
	4	0.8393	4	Vs = 155
Minimize System Weight	1	0.9668	2	Rs = 0.9769
	2	0.8715	2	Cs = 440.5520
	3	0.9572	2	Ws = 89.0309
	4	0.9382	2	Vs = 32
Multi-objective Functions	1	0.8536	3	Rs = 0.9757
	2	0.7977	3	Cs = 114.0175
	3	0.9189	2	Ws = 147.0485
	4	0.8133	3	Vs = 57

Table 5. Simulation results using fmincon optimization method when system reliability $R_s = A$.

Objective	Stage	Reliability	Component	Simulation Result
Minimize System Cost	1	0.5846	5	Rs = 0.9500
	2	0.5184	6	Cs = 36.0616
	3	0.6988	4	Ws = 475.1981
	4	0.5252	5	Vs = 195
Minimize System Weight	1	0.9534	1	Rs = 0.9500
	2	0.9313	2	Cs = 422.7688
	3	0.9770	1	Ws = 60.8431
	4	0.9351	2	Vs = 20
Multi-objective Functions (Cost + Weight)	1	0.8326	3	Rs = 0.9500
	2	0.7755	3	Cs = 91.7003
	3	0.9053	2	Ws = 147.0485
	4	0.7840	3	Vs = 57

Table 6. Simulation results using fminimax optimization method when system reliability $R_s = A$.

Objective	Stage	Reliability	Component	Simulation Result
Minimize System Cost	1	0.5846	5	Rs = 0.9500
	2	0.5184	6	Cs = 36.0616
	3	0.6988	4	Ws = 475.1981
	4	0.5252	5	Vs = 195
Minimize System Weight	1	0.9534	1	Rs = 0.9500
	2	0.9313	2	Cs = 422.7688
	3	0.9770	1	Ws = 60.8431
	4	0.9351	2	Vs = 20
Multi-objective Functions (Cost + weight)	1	0.8325	3	Rs = 0.9500
	2	0.7755	3	Cs = 91.7228
	3	0.9054	2	Ws = 147.0485
	4	0.7841	3	Vs = 57

Table 7. Simulation results using GA optimization method when system reliability $R_s = A$.

Objective	Stage	Reliability	Component	Simulation Result
Minimize System Cost	1	0.6317	5	Rs = 0.9500
	2	0.5327	5	Cs = 37.5962
	3	0.6800	5	Ws = 425.1462
	4	0.5980	4	Vs = 182

Continued

Minimize System Weight	1	0.8925	2	$R_s = 0.9500$
	2	0.9369	2	$C_s = 426.6653$
	3	0.9720	1	$W_s = 72.9236$
	4	0.9440	2	$V_s = 23$
Multi-objective Functions (Cost + Weight)	1	0.8179	3	$R_s = 0.9500$
	2	0.7812	3	$C_s = 83.8740$
	3	0.8894	2	$W_s = 47.0485$
	4	0.7656	3	$V_s = 57$

Table 8. Simulation results using hybrid optimization method when system reliability $R_s = A$.

Objective	Stage	Reliability	Component	Simulation Result
Minimize System Cost	1	0.5846	5	$R_s = 0.9500$
	2	0.5184	6	$C_s = 36.0616$
	3	0.6988	4	$W_s = 475.1981$
	4	0.5252	5	$V_s = 195$
Minimize System Weight	1	0.9534	1	$R_s = 0.9500$
	2	0.9313	2	$C_s = 422.7688$
	3	0.9770	1	$W_s = 60.8431$
	4	0.9351	2	$V_s = 20$
Multi-objective Func- tions (Cost + Weight)	1	0.8326	3	$R_s = 0.9500$
	2	0.7755	3	$C_s = 91.7003$
	3	0.9053	2	$W_s = 147.0485$
	4	0.7840	3	$V_s = 57$

6. Hybrid Genetic Algorithm (HGA) for Multi-Objective Optimization

Most previous studies have focused on several methods for solving redundancy optimization problems. In this study, we develop an approach by considering some aspects that have not been considered previously. The mathematical model represents the multi-objective HGA with a constraint-handling strategy for solving the proposed model. HGA is a meta-heuristic method that is used to solve optimization problems efficiently. In this method, first, an initial set of random potential solutions including a number of particles is created. Each particle represents a solution of the problem and has a position and velocity that change in each iteration so that better solutions can be obtained.

7. A Case Study: Overspeed Protection System for a Gas Turbine

To evaluate the performance of the HGA in reliability optimization problems, overspeed detection continuously provided by the electrical and mechanical systems is considered in a case study. The benchmark considered is an overspeed protection system for a gas turbine. When overspeed occurs, it is necessary to cut off the fuel supply using control valves, *i.e.*, the four valve controllers (V1-V4) must close. The control system is modeled as a four-stage series-parallel system, as shown in **Figure 3**.

Each stage represents a controller that can be considered as a parallel system. All the components of the system have the same failure rate. The equivalent circuit of the overspeed control system is shown in **Figure 4**.

Here, v_i is the volume of each component in subsystem i , V is the upper limit on the sum of the subsystem products of volume and weight, C is the upper limit on the system cost, and W is the upper limit on the system weight. The parameters α_i and β_i are constants representing the physical characteristics of each component in stage i . T is the operating time during which a component must not fail. The input parameters of the overspeed protection system for a gas turbine are listed in **Table 9**.

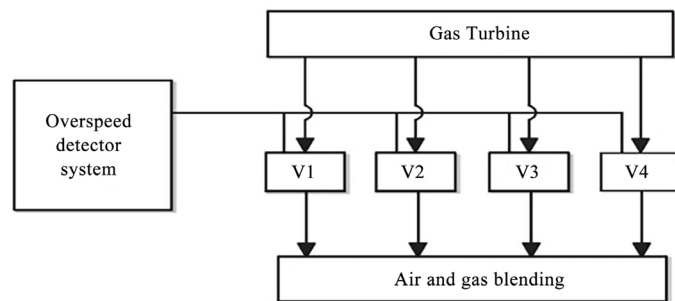


Figure 3. Block diagram of overspeed protection system for gas turbine with four valves.

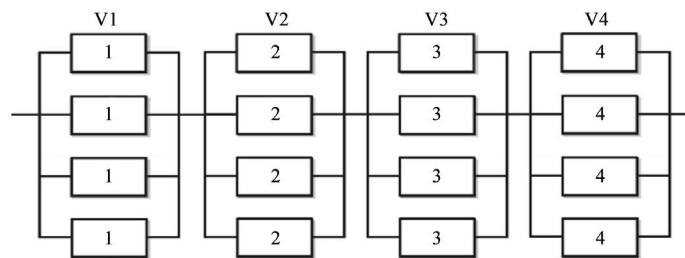


Figure 4. Equivalent circuit: four-stage series-parallel system.

Table 9. Design values of different parameters used in overspeed protection system of gas turbine.

Number of stages		4		
Lower limit on R_s		0.95		
Upper limit on cost		400		
Upper limit on weight		500		
Upper limit on volume		250		
Operating time		1000 hours		
Stage	$10^5 \alpha_i$	β_i	v_i	w_i
1	1.0	1.5	1	6
2	2.3	1.5	2	6
3	0.3	1.5	3	8
4	2.3	1.5	2	7

8. Computational Results and Discussion

We compared our solutions with those obtained in a previous study [23]. From **Table 10**, it is clear that our HGA approach obtains better solutions for the series-parallel system compared to the other approaches presented in the literature. The best fitness and mean fitness of the system cost, system weight, and multi-objective functions are shown in **Figures 5-7**, respectively.

The mathematical model used for calculating the objective function is employed to define the solution that guarantees an optimal trade-off between the two objectives, and the result is shown in **Figure 8**. **Figure 9** shows the average distance between individuals.

These figures show the number of generations in GA. In addition, the values of each objective function in each iteration are shown. The toolbox is employed to generate these figures, which can be used to determine the most suitable reliability level that minimizes the total cost, weight, and volume subject to various constraints.

The runs of the HGA were continuously monitored throughout the generations (**Figures 5-7**). These plots show the best and mean fitness values of the fitness functions after 100, 100, and 300 generations, respectively. For **Figure 5**, the best fitness is in the range of 38.787 and the mean fitness is in the range of 38.795. For **Figure 6**, the best fitness is in the range of 55.9112 and the mean fitness is in the range of 55.9136. For **Figure 7**, the best fitness is in the range of 0.462836 and the mean fitness is in the range of 0.462929. From these plots, it can easily be observed that the fitness value converges toward the optimal value from generation to generation.

Table 10. Comparison of simulation results of optimal solutions of single- and multi-objective function for series-parallel system using HGA with other results presented in the literature.

Objective	Stage	Results given in Ref. [23]			Results given by hybrid genetic algorithm		
		Reliability	Component	Simulation result	Reliability	Component	Simulation result
Maximize System Reliability	1	0.866288	6.0	$R_s = 0.999881$	0.8971	5	$R_s = 0.9999$
	2	0.850029	6.0	$C_s = 381.12183$	0.8659	6	$C_s = 381.5582$
	3	0.918417	4.0	$W_s = 485.77850$	0.9358	4	$W_s = 475.1981$
	4	0.913049	4.0	$V_s = 188.0$	0.8769	5	$V_s = 195$
Minimize System Cost	1	0.559777	6.0	$R_s = 0.971340$	0.7997	4	$R_s = 0.9939$
	2	0.599392	6.0	$C_s = 54.472889$	0.7896	4	$C_s = 133.4582$
	3	0.685273	4.0	$W_s = 485.778504$	0.7154	5	$W_s = 346.2031$
	4	0.703375	4.0	$V_s = 188.0$	0.8393	4	$V_s = 155$
Minimize System Weight	1	0.864883	3.0	$R_s = 0.971597$	0.9668	2	$R_s = 0.9769$
	2	0.944821	2.0	$C_s = 295.029388$	0.8715	2	$C_s = 440.5520$
	3	0.905934	2.0	$W_s = 107.352295$	0.9572	2	$W_s = 89.0309$
	4	0.880399	2.0	$V_s = 370$	0.9382	2	$V_s = 32$
Multi-Objective Optimization	1	0.820009	4.0	$R_s = 0.971641$	0.8536	3	$R_s = 0.9757$
	2	0.806433	3.0	$C_s = 119.04067$	0.7977	3	$C_s = 114.0175$
	3	0.869349	3.0	$W_s = 177.234863$	0.9189	2	$W_s = 147.0485$
	4	0.865680	2.0	$V_s = 69.0$	0.8133	3	$V_s = 57$

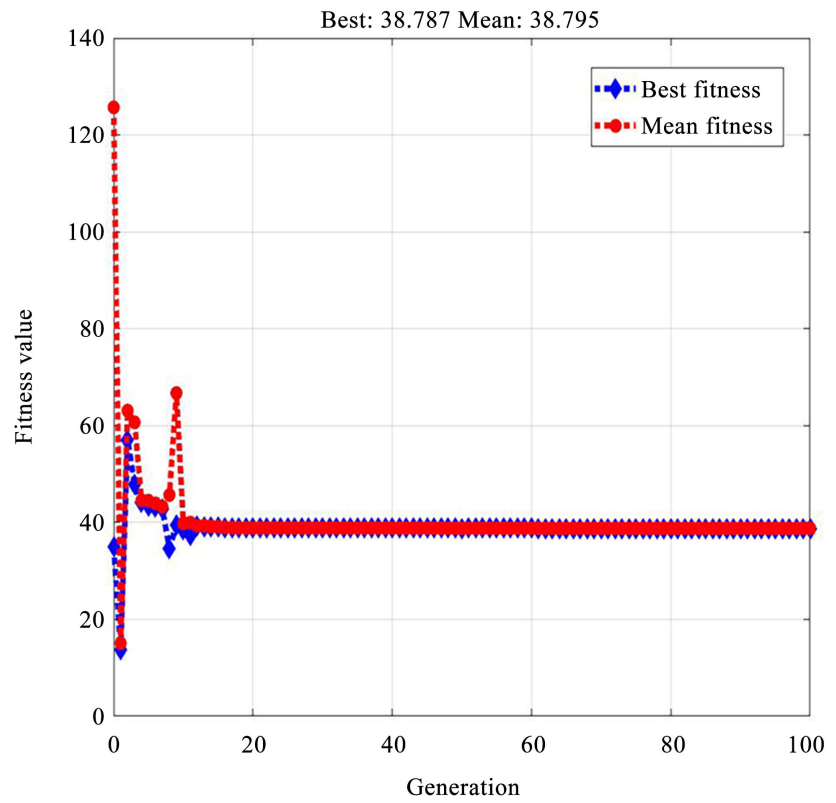


Figure 5. Best fitness and mean fitness of the system cost.

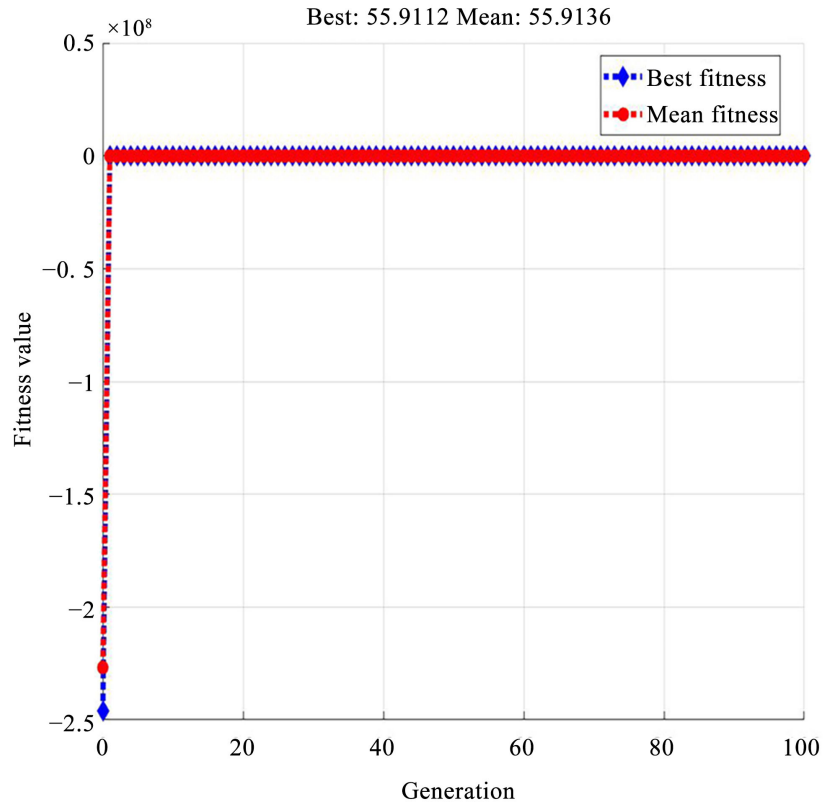


Figure 6. Best fitness and mean fitness of the system weight.

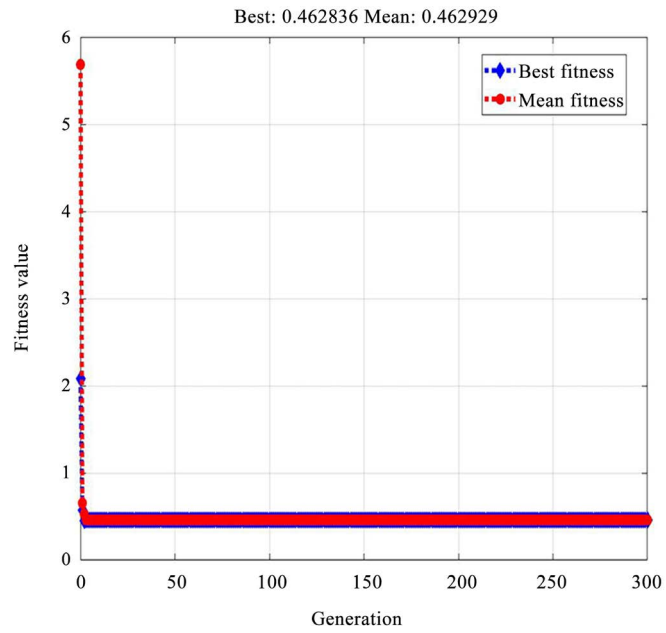


Figure 7. Best fitness and mean fitness of the multi-objective functions.

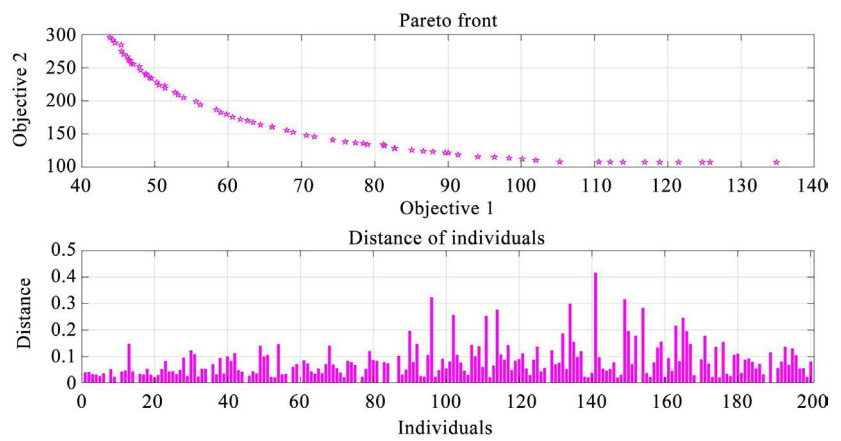


Figure 8. Overall best Pareto front obtained by multi-objective optimization and HGA: cost vs. weight and distance of individuals.

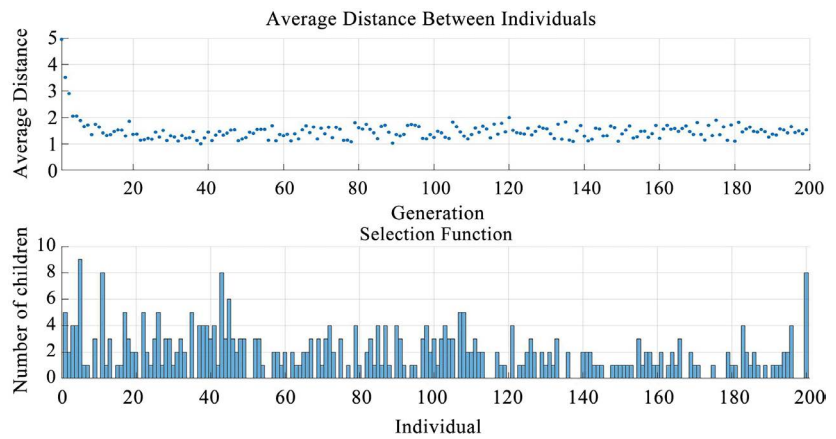


Figure 9. Average distance between individuals.

The upper plot function in **Figure 8** is the HGA Pareto function, which plots the Pareto front (limited to any three objectives) at every generation. This plot shows the trade-off between the two components of f . It is plotted in the objective function space. The lower plot shows the histogram distances of individuals.

The upper plot in **Figure 9** shows the average distance between individuals for each objective, which is a good measure of the diversity of the initial population that affects the performance of the HGA. In general, if the diversity is too high or too low, the HGA might not perform well. Here, it is obvious that the distance does not reach extreme values, so it is considered that the performance is good. The lower plot shows the histogram of the parents, which indicates the parents that contribute to each generation of children populated by each individual.

We considered only the case of multi-objective optimization with the HGA technique for our contribution, and we generated/calculated the values of Ws , Cs , and Vs for 19 values of $Rs = A$ ($A = 0.9900, 0.9905, 0.9910, 0.9915, \dots, 0.9980, 0.9985, \text{ and } 0.9990$). The results are summarized in **Table 11(a)**. On the basis of these tables, we plotted the curves r_i for each stage and Cs , Ws , and Vs as functions of Rs . Further, we determined the mathematical equation of each of these curves. We used the nonlinear regression technique. If the utility of these equations is good, we can use them to estimate/calculate the values of $r_1, r_2, r_3, r_4, Cs, Ws$, and Vs for any value of Rs ($Rs = 0.9900$ to 0.9990). Thus, from a practical point of view, these equations are extremely useful. We also obtained the nonlinear regression fitted line plot, which can be used to investigate the relationship between two continuous variables, namely a response variable and a predictor variable. Thus, we can derive a regression equation and plot the regression line. For the copper expansion data, the method determines the type of relationship with these graphs and the line is fitted as per the requirement of data points. Minitab uses the Gauss-Newton algorithm, imposes a maximum of 200 iterations, and employs a tolerance of 0.00001 to achieve convergence. It displays a plot of the data overlaid with a curve illustrating the best-fitting equation based on our expectation function. The plot of the copper expansion data indicates that the specified rational polynomial is a good fit for the data. The points are fairly close to the curve and follow the curve without any systematic deviations from it. If we fit these models, differences will be observed in the desired values as well as in the corresponding points in the graph. This is because the fitted value is given, not the original one. Therefore, it is called the expected value or return of the model. The values of the parameters can be obtained using Minitab. We simply put the values of the parameters in the following regression equation.

$$\text{Explained variable} = a + b * Rs + c * Rs^2 + d * Rs^3.$$

With the parameter estimates in **Table 12**, we obtain $r_{1New}, r_{2New}, r_{3New}, r_{4New}, CsNew, WsNew$, and $VsNew$, as with each value of Rs obtained previously. Then, we obtain a scatter plot between Rs and $r_{1New}, r_{2New}, r_{3New}, r_{4New}, CsNew, WsNew$, and $VsNew$, as shown in **Figure 10**. There will be

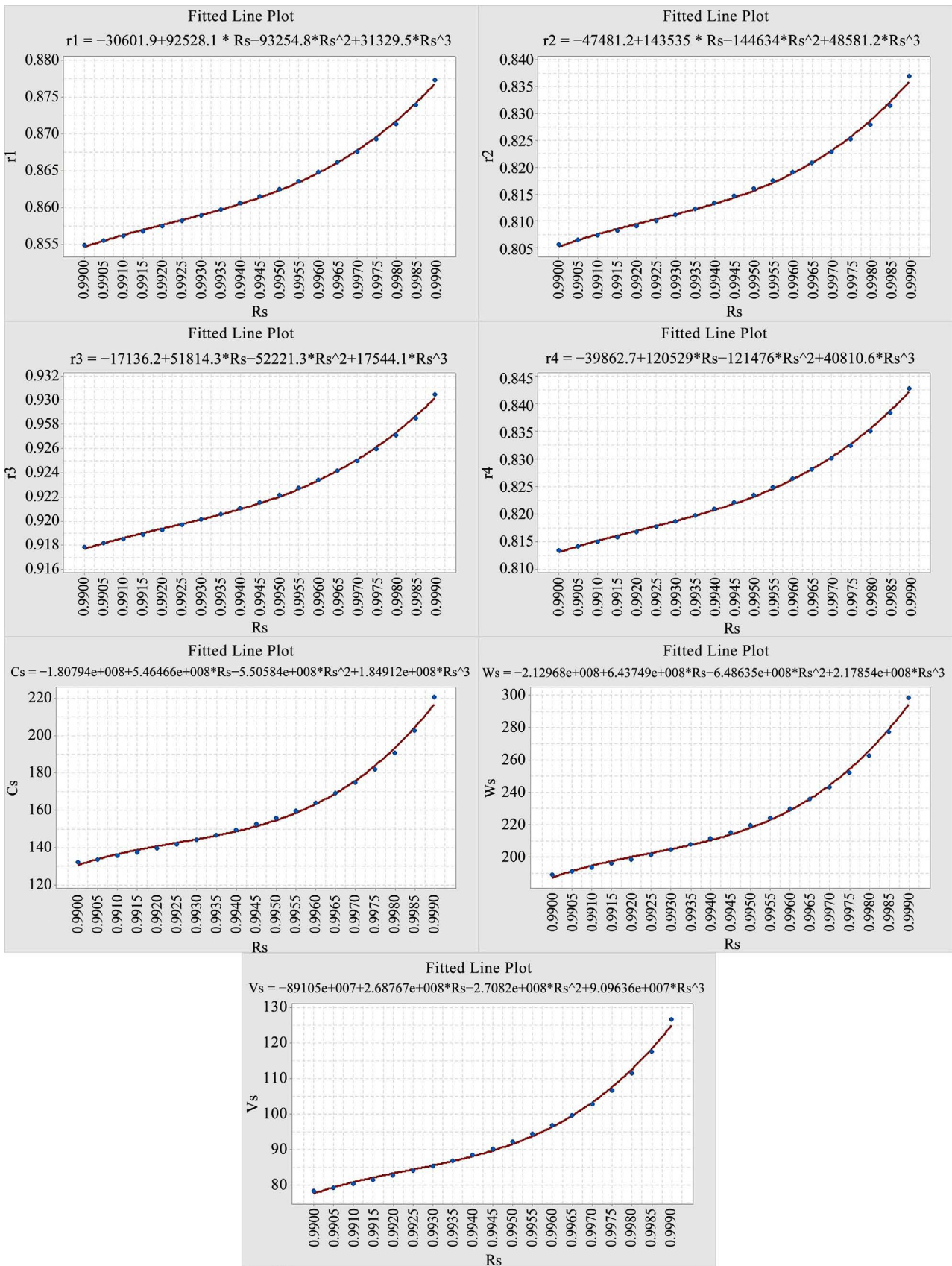


Figure 10. Scatter plot of $r_1, r_2, r_3, r_4, C_s, W_s,$ and V_s vs. R_s - (R_s) = 0.9900 - 0.9990.

Table 11. (a) Optimum solutions of HGA for multi-objective optimizations when $R_s = 0.990 - 0.9990$; (b) Optimum solutions of HGA for multi-objective optimizations when $R_s = 0.990 - 0.9990$ after approximating the values of r_i to four decimals places and adjusting the values of m_i to integer values.

(a)

	Reliability r_i for the stage (1, 2, 3, 4)				Component m_i for the stage (1, 2, 3, 4)				Simulation Result			
	r_1	r_2	r_3	r_4	m_1	m_2	m_3	m_4	C_s	W_s	V_s	
0.9900	0.85487685	0.805655761	0.917846852	0.81339754	3.190169321	3.530179896	2.587664286	3.406172744	132.164455	189.0930845	78.39356539	
0.9905	0.855473088	0.806463951	0.918182363	0.814181873	3.210073417	3.552423421	2.604289554	3.428104828	133.858123	191.2699882	79.39477332	
0.9910	0.856095986	0.807307881	0.918532889	0.815000666	3.230988456	3.575790604	2.62176897	3.451147483	135.6554486	193.5761858	80.45369878	
0.9915	0.856748082	0.808191244	0.918899784	0.815857538	3.253032189	3.600408495	2.640202236	3.475425385	137.5689444	196.0275166	81.57726785	
0.9920	0.857432193	0.809118161	0.919284538	0.816756654	3.276345812	3.626430945	2.659708108	3.501087952	139.6133923	198.6430896	82.77382002	
0.9925	0.858152802	0.810094223	0.919689825	0.817703212	3.301079021	3.654026461	2.68041401	3.528304218	141.8071232	201.4443344	84.05266056	
0.9930	0.858914505	0.811125576	0.920118304	0.81870315	3.327424121	3.683409522	2.702481827	3.557285894	144.1720581	204.4583748	85.42555264	
0.9935	0.85972226	0.812219748	0.920572273	0.819763928	3.355626654	3.714840684	2.72612569	3.588288077	146.7349448	207.7191939	86.90721932	
0.9940	0.860583831	0.813386219	0.921056565	0.820894404	3.385961833	3.748634381	2.751574935	3.621625918	149.5299775	211.2673301	88.51509943	
0.9945	0.86150696	0.814636401	0.921575347	0.822105855	3.418809142	3.785198577	2.779150387	3.657697193	152.5998752	215.1564426	90.27224062	
0.9950	0.862503387	0.815985304	0.92213516	0.823412564	3.454621766	3.825042849	2.80924355	3.697009653	156.0008801	219.454241	92.20762586	
0.9955	0.863587167	0.817452102	0.922744026	0.8248332	3.494016405	3.8688423	2.842375881	3.740227437	159.8074131	224.2515769	94.35993664	
0.9960	0.864776784	0.819062299	0.923412064	0.826392467	3.537830595	3.917512894	2.87926149	3.788253165	164.1211822	229.6735445	96.78222413	
0.9965	0.866099269	0.820851941	0.924154577	0.828124752	3.587206353	3.972314727	2.920877079	3.842338221	169.0876248	235.895234	99.54831274	
0.9970	0.867592417	0.822871926	0.924992706	0.830079694	3.643828911	4.035101433	2.968660812	3.904307868	174.9221985	243.1777639	102.7676572	
0.9975	0.869314318	0.825200833	0.925958999	0.832332426	3.710287367	4.108711835	3.024826368	3.976975743	181.9651363	251.930604	106.610654	
0.9980	0.871358752	0.827965665	0.927105963	0.8350061	3.790915108	4.197894565	3.093078877	4.065023275	190.7966515	262.8530903	111.3659142	
0.9985	0.873899579	0.831400585	0.928530491	0.838326142	3.89372908	4.311439621	3.180304469	4.177144828	202.5337544	277.2792866	117.5782367	
0.9990	0.877312537	0.836996655	0.930443934	0.84278106	4.036567983	4.468847795	3.301807032	4.332627137	220.4812934	298.2808393	126.4841872	

(b)

R_s	Reliability r_i for the stage (1, 2, 3, 4)				Component n_i				Simulation Result		
	r_1	r_2	r_3	r_4	n_1	n_2	n_3	n_4	C_s	W_s	V_s
0.9900	0.8549	0.8057	0.9178	0.8134	3	4	3	3	133.7041	198.6198	86
0.9905	0.8555	0.8065	0.9182	0.8142	3	4	3	3	134.6502	198.6198	86
0.9910	0.8561	0.8073	0.9185	0.8150	3	4	3	3	135.5692	198.6198	86
0.9915	0.8567	0.8082	0.9189	0.8159	3	4	3	3	136.6158	198.6198	86
0.9920	0.8574	0.8091	0.9193	0.8168	3	4	3	4	150.4994	230.2647	100
0.9925	0.8582	0.8101	0.9197	0.8177	3	4	3	4	151.7831	230.2647	100
0.9930	0.8589	0.8111	0.9201	0.8187	3	4	3	4	153.1025	230.2647	100
0.9935	0.8597	0.8122	0.9206	0.8198	3	4	3	4	154.608	230.2647	100
0.9940	0.8606	0.8134	0.9211	0.8209	3	4	3	4	156.2153	230.2647	100
0.9945	0.8615	0.8146	0.9216	0.8221	3	4	3	4	157.9011	230.2647	100
0.9950	0.8625	0.8160	0.9221	0.8234	3	4	3	4	159.7984	230.2647	100
0.9955	0.8636	0.8175	0.9227	0.8248	3	4	3	4	161.912	230.2647	100
0.9960	0.8648	0.8191	0.9234	0.8264	4	4	3	4	173.4566	257.3974	107
0.9965	0.8661	0.8209	0.9242	0.8281	4	4	3	4	176.295	257.3974	107
0.9970	0.8676	0.8229	0.9250	0.8301	4	4	3	4	179.5854	257.3974	107
0.9975	0.8693	0.8252	0.9260	0.8323	4	4	3	4	183.4539	257.3974	107
0.9980	0.8714	0.8280	0.9271	0.8350	4	4	3	4	188.3124	257.3974	107
0.9985	0.8739	0.8314	0.9285	0.8383	4	4	3	4	194.5076	257.3974	107
0.9990	0.8773	0.8370	0.9304	0.8428	4	4	3	4	204.0998	257.3974	107

Table 12. Explained variable with parameters when $R_s = 0.9900 - 0.9990$.

Explained variable	r_1N	r_2N	r_3N	r_4N	C_sN	W_sN	V_sN
Parameter							
a	-30601.9	-47481.2	-17136.2	-39862.7	-180794266	-212968023	-88910519
b	92528.1	143535	51814.3	120529	546466156	643748903	268766970
c	-93254.8	-144634	-52221.3	-121476	-550583510	-648634663	-270819890
d	31329.5	48581.2	17544.1	40810.6	184911867	217854114	90963580

non-linear parameters when we fit the models given previously.

The results obtained using multi-objective optimizations with the HGA are summarized in **Table 11(a)**. It can be seen that the number of components n_i and the individual component reliability r_i in various stages are different. However, in practice, n_i must be an integer. Therefore, must approximate the values of r_i and adjust the values of n_i to integer values. The new results with this approximation are summarized in **Table 11(b)**. It can be seen that when the num-

bers of components in different stages, n_p , are modified to integer values, the cost, weight, and volume of the system are reduced slightly.

In this study, we must ensure that the number of simulations (n) for each time is sufficient to achieve convergence. To this end, we changed the value of n (0, 1, 2, 3, ..., 75), and for the simulation with different values of R_s , we can say that for all values of R_s , the simulation process converges at $n = 30$, and for the case of $R_s = 0.9900$, the converged values of r_1 , r_2 , r_3 , and r_4 are 0.8724, 0.9567, 0.8838, and 0.8668, respectively, as shown in **Figure 11**. In summary, we have discussed our novel approach, *i.e.*, design of system reliability using the simulation process. The advantage of our approach is that the reliable regression curves have been generated using the proposed simulation process (**Figure 10**) and the utility of these curves for the system design is that they can help the designer to determine any level of reliability r_i of the system components, the corresponding value of cost, weight, and volume depending on the chosen value of R_s .

9. Conclusions

In this study, we proposed a hybrid genetic algorithm and presented a novel system design for the entire system with the desired level of reliability. Thus, we achieved two objectives. First, we evaluated our approach to determine the robustness of our method by comparing it with another method in the literature. The results indicated that our approach yields better results. Second, we used this approach to develop a new simulation process for system design. We varied R_s and obtained different r_1 , r_2 , r_3 , r_4 , C_s , W_s , and V_s . Then, we plotted the curves, which are of great practical significance because they enable the designer of the system to determine the values of r_1 , r_2 , r_3 , r_4 , C_s , W_s , and V_s corresponding to the value of R_s . Using $R_s = 0.9904$, the designer could directly use the curves to obtain all the required values. Some values converge after several iterations in some cases. The performance and robustness of the proposed approach can easily be evaluated. Rapid convergence can be achieved using our model and approach, as shown in **Figure 11**. Moreover, robustness can be confirmed on the basis of similar results obtained under different initial conditions, as shown in **Figure 11**. In addition, **Figure 10** illustrates the practical utility of our approach, *i.e.*, the designer can determine the reliability of each component corresponding to any value of system reliability R_s .

Finally, we fixed the system reliability to obtain a satisfactory system with minimum cost and weight. Comparison of the simulation results indicates the superiority of HGA over other algorithms in terms of searching quality and robustness of the solution. The main advantage of the proposed multi-objective approach is that it offers greater flexibility to system designers for testing problems. Our HGA improves the objective function values and gives the best-known solutions for benchmark suites. Thus, to the best of our knowledge, HGA is an effective algorithm for application to the RRAP. It is especially useful when the optimization problem under consideration is complex.

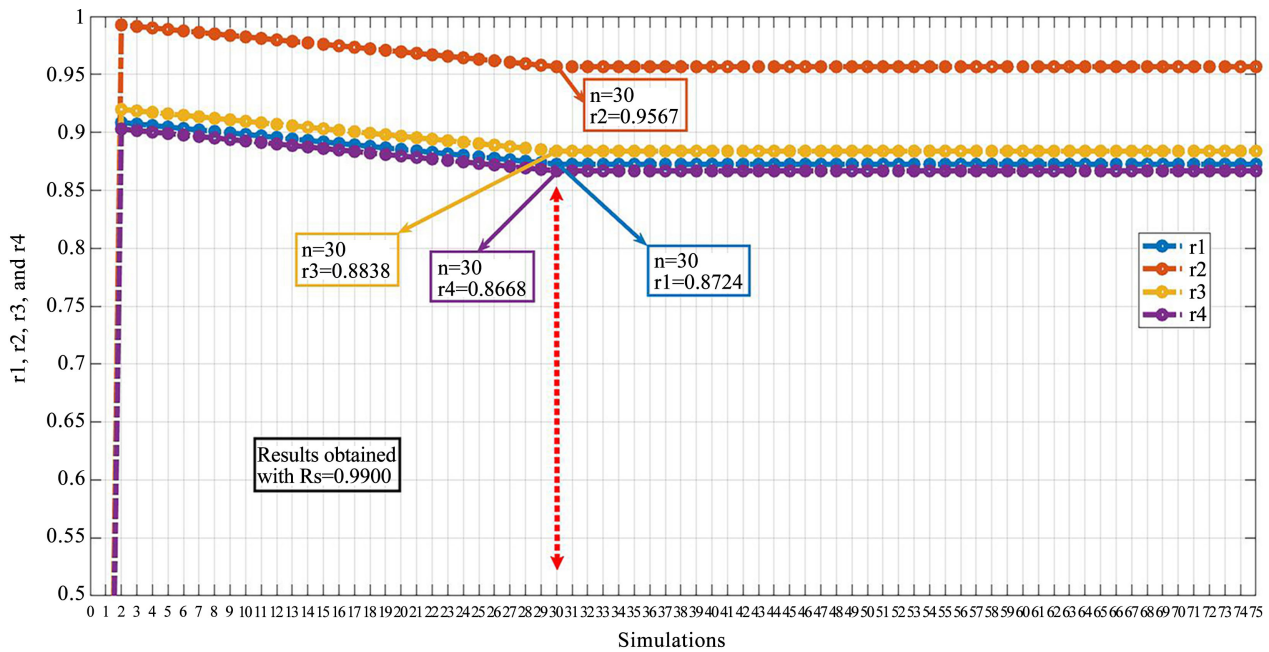


Figure 11. Scatter plot of r_1 , r_2 , r_3 , and r_4 versus number of simulations (n).

In the future, we will focus on extending our approach to other algorithms, such as hybrid nonlinear mixed integer programming, to achieve better results.

References

- [1] Kuo, W. and Wan, R. (2007) Recent Advances in Optimal Reliability Allocation. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, **37**, 143-156. <https://doi.org/10.1109/TSMCA.2006.889476>
- [2] Kuo, W. and Prasad, V.R. (2000) An Annotated Overview of System-Reliability Optimization. *IEEE Transactions on Reliability*, **49**, 176-187. <https://doi.org/10.1109/24.877336>
- [3] Kuo, W., Hwang, C.L. and Tillman, F.A. (1978) A Note on Heuristic Methods in Optimal System Reliability. *IEEE Transactions on Reliability*, **27**, 320-324. <https://doi.org/10.1109/TR.1978.5220401>
- [4] Forsthofer, W.E. (2005) Forsthofer's Rotating Equipment Handbooks: Fundamentals of Rotating Equipment. Elsevier, New York.
- [5] Kundur, P. (1994) Power System Stability and Control. McGraw-Hill, New York.
- [6] Hejzlar, P., Ubra, O. and Ambroz, J. (1993) A Computer Program for Transient Analysis of Steam Turbine Generator Overspeed. *Nuclear Engineering and Design*, **144**, 469-485. [https://doi.org/10.1016/0029-5493\(93\)90043-9](https://doi.org/10.1016/0029-5493(93)90043-9)
- [7] Seebregts, A.J., Rademakers, L.W.M.M. and Van Den Horn, B.A. (1995) Reliability Analysis in Wind Turbine Engineering. *Microelectronics Reliability*, **35**, 1285-1307. [https://doi.org/10.1016/0026-2714\(95\)99378-V](https://doi.org/10.1016/0026-2714(95)99378-V)
- [8] Dhingra, A.K. (1992) Optimal Apportionment of Reliability and Redundancy in Series Systems under Multiple Objectives. *IEEE Transactions on Reliability*, **41**, 576-582. <https://doi.org/10.1109/24.249589>
- [9] Fetanat, A., Shafipour, G. and Ghanatir, F. (2012) Reliability Redundancy Optimization for Optimal Designing in Gas Turbines' Overspeed Protection Using Elitism Box-Muller Harmony Search Algorithm. *Journal of Theoretical and Applied Infor-*

- mation Technology*, **46**, 991-1000.
- [10] Rao, S.S. and Dhingra, A.K. (1992) Reliability and Redundancy Apportionment Using Crisp and Fuzzy Multiobjective Optimization Approaches. *Reliability Engineering & System Safety*, **37**, 253-261.
[https://doi.org/10.1016/0951-8320\(92\)90131-4](https://doi.org/10.1016/0951-8320(92)90131-4)
- [11] Luus, R. (1975) Optimization of System Reliability by a New Nonlinear Integer Programming Procedure. *IEEE Transactions on Reliability*, **24**, 14-16.
<https://doi.org/10.1109/TR.1975.5215316>
- [12] Hikita, M., Nakagawa, Y., Nakashima, K. and Narihisa, H. (1992) Reliability Optimization of Systems by a Surrogate-Constraints Algorithm. *IEEE Transactions on Reliability*, **41**, 473-480. <https://doi.org/10.1109/24.159825>
- [13] Gopal, K., Aggarwal, K.K. and Gupta, J.S. (1978) An Improved Algorithm for Reliability Optimization. *IEEE Transactions on Reliability*, **27**, 325-328.
<https://doi.org/10.1109/TR.1978.5220403>
- [14] Takao, Y., Mitsuo, G., Yinxiu, L. and Chang, E.K. (1996) A Genetic Algorithm for Interval Nonlinear Integer Programming Problem. *Computers & Industrial Engineering*, **31**, 913-917.
- [15] Hsieh, Y.C., Chen, T.C. and Bricker, D.L. (1998) Genetic Algorithms for Reliability Design Problems. *Microelectronics Reliability*, **38**, 1599-1605.
[https://doi.org/10.1016/S0026-2714\(98\)00028-6](https://doi.org/10.1016/S0026-2714(98)00028-6)
- [16] Zhao, J.Q., Wang, L., Zeng, P. and Fan, W.H. (2012) An Effective Hybrid Genetic Algorithm with Flexible Allowance Technique for Constrained Engineering Design Optimization. *Expert Systems with Applications*, **39**, 6041-6051.
<https://doi.org/10.1016/j.eswa.2011.12.012>
- [17] Kanagaraj, G., Ponnambalam, S.G. and Jawahar, N. (2013) A Hybrid Cuckoo Search and Genetic Algorithm for Reliability-Redundancy Allocation Problems. *Computers & Industrial Engineering*, **66**, 1115-1124.
<https://doi.org/10.1016/j.cie.2013.08.003>
- [18] Ghodrati, A. and Lotfi, S. (2012) A Hybrid CS/GA Algorithm for Global Optimization. *Proceedings of the International Conference on Soft Computing for Problem Solving*, Roorkee, 20-22 December 2011, 397-404.
https://doi.org/10.1007/978-81-322-0487-9_38
- [19] Gen, M. and Yun, Y. (2006) Soft Computing Approach for Reliability Optimization: State-of-the-Art Survey. *Reliability Engineering & System Safety*, **91**, 1008-1026.
<https://doi.org/10.1016/j.res.2005.11.053>
- [20] Zou, D., Gao, L., Li, S. and Wu, J. (2011) An Effective Global Harmony Search Algorithm for Reliability Problems. *Expert Systems with Applications*, **38**, 4642-4648.
<https://doi.org/10.1016/j.eswa.2010.09.120>
- [21] Abouei Ardakan, M., Sima, M., Zeinal Hamadani, A. and Coit, D.W. (2016) A Novel Strategy for Redundant Components in Reliability—Redundancy Allocation Problems. *IIE Transactions*, **48**, 1043-1057.
<https://doi.org/10.1080/0740817X.2016.1189631>
- [22] Afonso, L.D., Mariani, V.C. and dos Santos Coelho, L. (2013) Modified Imperialist Competitive Algorithm Based on Attraction and Repulsion Concepts for Reliability-Redundancy Optimization. *Expert Systems with Applications*, **40**, 3794-3802.
<https://doi.org/10.1016/j.eswa.2012.12.093>
- [23] Quy, N. (1998) Reliability Optimization in Mechanical System Design by Fuzzy Logic. Master's Thesis, École de Technologie Supérieure, Université du Québec, Québec.

Abbreviations and Acronyms

MOO: Multi-objective optimization; HGA: Hybrid genetic algorithm; GA: Genetic algorithm; RAP: Redundancy allocation problem; RRAP: Reliability-redundancy allocation problem; HAS: Harmony search algorithm; EBMHSA: Elitism Box-Muller harmony search algorithm; MVGA: Modified version of the genetic algorithm; ICA: Imperialist competitive algorithm; Fmincon: Find minimum of constrained; Fminimax: Solve minimax constraint problem.