Multi-Task Optimization in Reliability Redundancy Allocation Problem: A Multifactorial Evolutionary-Based Approach

Md. Abdul Malek Chowdury[1], Rahul Nath[2], Amit K. Shukla[3], Amit Rauniyar[4], Pranab K. Muhuri[1]

[1]Department of Computer Science, *South Asian University*, New Delhi-110021, India

[2]Department of Informatics, *University of Bergen*, Thormøhlens Gate 55, 5008 Bergen, Norway

[3]School of Technology and Innovations, *University of Vaasa*, Wolffintie 34, FI-65200 Vaasa, Finland

[4]School of Computer Science, *University College Dublin*, Dublin 04, Ireland

Email: {mdamalekchy@gmail.com; rahul.nath@uib.no; amit.shukla@uwasa.fi; amit.rauniyar@ucd.ie; pranabmuhuri@cs.sau.ac.in}

Abstract— Evolutionary multi-task optimization attempts to solve multiple optimization problems simultaneously by modeling the solution structures of two or more problems within a single encoding. In this paper, we report a novel way for evolutionary multi-task optimization in the reliability redundancy allocation problem exploiting the concepts of the popular multifactorial evolutionary algorithm (MFEA). We demonstrate the working of the proposed method considering two test sets and show how they can be concurrently solved using the MFEA. In the first test set, we consider two optimization tasks (case studies): the complex (bridge) system and the series-parallel system. In the second test set, there are two optimization tasks: the over-speed protection system for the gas turbine and the life support system in a space capsule. The common attributes between the two systems, within a set, complement each other to enhance the evolution process through implicit knowledge transfer. We present the comparative results considering existing evolutionary methods such as particle swarm optimization, genetic algorithm, simulated annealing, differential evolution, and ant colony optimization. Results are analyzed and compared with other approaches, viz., PSO, GA, SA, DE and ACO, using the average reliability, best reliability, computation time, performance ranking, and the popular statistical significance test of analysis of variance (ANOVA). The outcome shows that our proposed approach can solve the multiple case studies of RRAP simultaneously without compromising the solution quality. Moreover, our MFEA based solution method tops the rank among all approaches and provides significant improvement in computation time where it gains 28.02% and 14.43% of

improvement in computation time for first and second test set, respectively, when compared with GA. The percentage improvements in the computational time of the MFEA significantly increases when it is compared with other approaches.

Keywords— *Evolutionary Multi-Task Optimization; Multifactorial optimization; Reliability-redundancy allocation problem (RRAP); Complex (Bridge) system; Series-parallel system; Over-speed protection system; Life support system in a space capsule.*

I. INTRODUCTION

In the realm of optimization algorithms, Evolutionary Multi-Task Optimization (EMTO) is a special type of multi-objective optimization (MOO) that utilizes the concept of multi-tasking in evolutionary algorithms (EAs). It attempts to solve multiple optimization problems simultaneously by modeling their solution structures within a single population. One of the widely used single-population EMTO approaches, which has been extensively utilized recently, is the multifactorial evolutionary algorithm (MFEA) [1]**.** Fig. 1 depicts the single/unified solution representations of MFEA while solving K number of Tasks. The task specific search space for Task-1, Task-2, and Task-k are denoted by $f_1(x), f_2(x)$ and $f_k(x)$, respectively. A unified solution (F) is formed by the combination of all task specific search space. Here, $x_1^*,\ x_2^* \dots, x_k^*$ represents the solution belonging to Task-1, Task-2 and Task-k, respectively. MFEA allows multiple individuals to evolve together which is good at different tasks. It enables these individual to learn together and enhance their performance in their assigned tasks [2]. Due to this feature, MFEA has shown significant performances in a wide range of applications and domains [3]**.** One of the complex problems where MFEA can be effectively used is reliability optimization.

Reliability of a system is defined as the probability that guarantees the failure-free/flawless functioning of an entire system or components associated with it for a particular time frame under the given environment. The system complexity in engineering applications such as transportation, communication, electrical power systems, etc., is proportional to the complexity of reliability optimization. In such cases, the system reliability becomes an essential requirement for ensuring flawless service. Hence, maximizing the system's reliability becomes a primary objective for a sustainable real-world application [4]. The reliability optimization falls under the umbrella of non-linear programming problems in which mainly three constraints: weight, volume, and cost, are associated. In order to improve the reliability of a system, two possible strategies can be considered which are mentioned as follows:

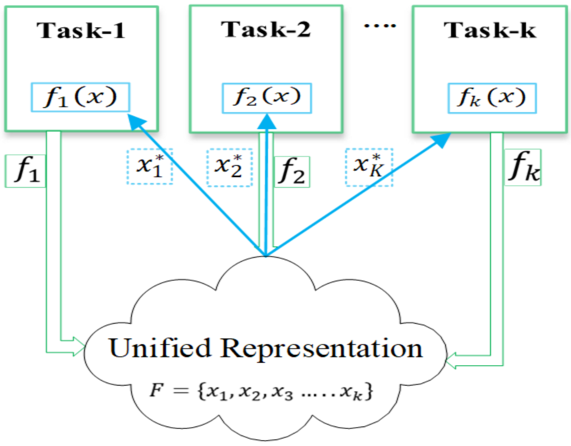


Fig. 1: Unified Representations of MFEA

a) The first strategy concentrates on optimizing the reliability of the components within the system under the constraints of the system resources. This strategy may increase the entire system's reliability until a certain level. Even though the component's reliability increases, achieving the desired system reliability sometimes becomes more challenging [5].

b) The second strategy involves adding redundant components to the different subsystems. These components when considered as a decision variable, converts the problem of optimizing the reliability of the entire system as the redundancy allocation problem (RAP) [6].

The combination of the above two strategies is called the reliability redundancy allocation problem (RRAP) [7]. The goal of the RRAP is to maximize the total system reliability by determining the optimum component's reliability and the number of redundant components. There are many structures and case studies for reliability optimization viz., series, parallel, series-parallel, complex (bridge) system, etc. [5], [8]. Over the years, RRAP has gained tremendous attraction by the research community to solve and address these cases using methods based on dynamic programming [9], Lagrangian multiplier [10], branch and bound, and linear programming [11]. Since RRAP is a non-linear optimization problem of the NP-hard category, the traditional approaches have limitations to solve the problem efficiently [12]. Therefore, several heuristic-based approaches have also been explored by many researchers to solve RRAP**.** However, most of the studies have attempted to optimize a single RRAP problem at a time [13]. There are no available approaches in the literature which has attempted to solve two cases of RRAP simultaneously. Due to this complex nature of the RRAP, we propose to use MFEA for such problems which might be well-suited to handle multiple tasks and optimize them simultaneously. Also, it will enable us to discover the trade-offs between systems sub-components to find the best solution.

In this paper, we report a novel approach of evolutionary multi-task optimization-based solution approach for the reliability redundancy allocation problem. The proposed approach has the ability to concurrently solve multiple case studies of RRAP by modeling their solution structures within a single encoding. It exploits the concepts of the popular multifactorial evolutionary algorithm. To demonstrate how two or more RRAP case studies (optimization tasks) can be concurrently solved using the multifactorial evolutionary algorithm, we consider two sets of case studies with different system structures. In the first test set, we consider two case studies: the complex (bridge) system and the series-parallel system, and then in the second test set also there are two case studies: the over-speed protection system for the gas turbine and the life support system in a space capsule. The common attributes between the system structures of two case studies within a test set complement each other to enhance the evolution process through implicit knowledge transfer. We have conducted extensive simulation experiments and comparative results are presented considering existing evolutionary methods such as particle swarm optimization, genetic algorithm, simulated annealing, differential evolution, and ant colony optimization. Thus, we show that our proposed approach has the ability to solve the multiple test cases of RRAP simultaneously without compromising the solution quality. We study and analyze the simulation results using average reliability for convergence, best reliability and computation time taken by each approach. Further we also conduct the TOPSIS method [14] to rank the algorithms on the basis of performance score and a statistical significance test of the results is performed using the popular ANOVA (analysis of variance) method.

The major points of our contributions and related experiments are as follows:

- It reports a novel approach of evolutionary multi-task optimization-based solution approach termed as MFEA (Multi-factorial evolutionary algorithm) for simultaneous optimization of the reliability redundancy allocation problem.
- We show the suitability of the proposed approach by conducting experiments on four benchmark problems of RRAP, which are formalized under two different test sets, which are then compared with five popular evolutionary approaches viz., PSO, GA, SA, DE and ACO.
- In order to establish the effectiveness of our proposed solution method in terms of solution quality and cost efficiency, we analyze the simulation results in terms of average reliability, best reliability and computation time.
- We also employ TOPSIS based MCDM technique using average & best reliability, and computation time to rank the algorithms on the basis of their performance.
- Further, we perform statistical significance test of the results using the popular ANOVA method.

Further, this paper is organized as follows: Section II provides the background study on evolutionary approaches for RRAP. Section III provides the details regarding the four RRAP case studies. Section IV is divided into three phases: in the first phase, an overview of MFEA algorithm is discussed; in second phase, discussion regarding the working of RRAP is covered, and the final phase covers the proposed solution approach and provides a detailed discussion on its important steps. Section V provides the details of the experiment results and comparative analysis for both the test sets. Section VI concludes the study with a discussion on the outcome of this work, the limitations of our proposed approach, and possible future work.

II. REVIEW OF RECENT EVOLUTIONARY BASED RRAP RESEARCH

In this section, we provide a concise review of the latest literature on evolutionary algorithms based RRAP research. A bi-objective simulation-based optimization algorithm was designed to solve RAP by Chambari et al. [15]. As an example of RAP, the series-parallel system with the state of active, cold, standby, mixed and k-mixed configuration was considered to solve using the popular NSGA-II. For the validation and effectiveness of the model, some benchmark problem was solved where the proposed approach outperformed with lower cost and improved reliability. A simulation sampling-based optimization model was modeled by Chambari et al. [16] to evaluate the reliability estimation through 4Dscript for tackling the difficulty of finding closed form in the single objective RAP. To find the optimal solution, GA was applied which showed the superiority in all the benchmark set solutions. Devi et al. [17] proposed a hybrid GA and PSO (HGAPSO) method to obtain system reliability of single objective series-parallel RAP problem. Authors conducted comparison between Heuristic Algorithm (HA), Constraint Optimization GA and HGAPSO on the basis of results obtained by system reliability and CPU time taken by these methods. Xu et al. [18] introduced a novel discrete bat algorithm to solve the problem of probabilistic common cause failures in heterogeneous RAP (series-parallel system) and Hamming-based bat movement, where Q learning based local search was adopted for better convergence. Modibbo et al. [19] presented a hybrid concept of estimation and optimization theory in RAP to help decision makers in heavily complex systems for estimation of system's component reliability and problem optimization for allocation and selection. Yeh [20] developed a BAT based algorithm namely bound-rule-BAT (BRB) for solving series-parallel RAP with mixed components where BRB algorithm provides an efficient pareto solution of RAP in comparison to the existing algorithm of simplified swarm optimization.

To obtain a higher reliability value, Dobani et al. [21] modeled a new heterogeneous RRAP problem by considering component mixing in existing RRAP problem. They solved the model using stochastic fractal search (SFS) which gives higher reliability and better structure. Muhuri et al. [22] have formulated the RRAP as a mixed integer bi-level optimization problem and proposed bi-level evolutionary algorithm based on quadratic approximations (BLEAQ) to maximize the reliability in the upper-level and minimize the total cost at lower-level. Dogahe et al. [23] proposed a new bi-objective model to optimize the RAP and reliability centralized maintenance (RCM) at the same time while considering component repair ability and non-reparability in a system. To find the trade-off between system reliability and cost, three meta-heuristics namely NSGA-II, MOPSO and Multi-objective firefly algorithm were applied and their comparison analysis was performed. Zand et al. [24] designed a SCADA water resource management control center using a bi-objective RAP which aims to simultaneously obtain maximum reliability with minimum cost. To address this problem, MOPSO algorithm was designed and performance was compared with epsilon-constrained method, where MOPSO demonstrated superiority in the results. Sharifi et al. [25] developed a RRAP for a series parallel problem with multi-state components. The main objective was to minimize the system cost under minimum reliability and allocating the optimal set of components with the consideration of failure probability of each component. They utilized immune algorithm to solve the developed problem. Filho et al. [26] applied a multi-objective linear approach for solving RRAP, where the earlier problem formulation was complex, non-linear and continuous. Afterward a linear programming technique namely CPLEX and an epsilon restricted multi-objective optimizer was applied to solve the presented formulation. In [27], Nath et al. proposed an approach to solve the RRAP considering all the objectives simultaneously using existing evolutionary approaches. Later, they extended the work in [28], where introduced a novel evolutionary approach to solve the RRAP by prioritizing the objectives. In [29], Alamdari et al. solved a joint availability-redundancy optimization problem using GA, SA, and an universal generating function, where components were considered as multi-state.

A HSGA was proposed by Garg et al. [30] to solve non-linear RAP which avoids individual weakness to ensure optimal reliability and less CPU time, compared to existing algorithms like HA, COGA, HPSO. Garg et al. [31] proposed a two-phase approach to solve RRAP. In first phase, an ACO algorithm was developed to find optimal reliability, while the second phase improved the solution by parameter free penalty technique. In [32], Guilani et al. developed an exact formulation based on Markov chain model to find optimal component sequences and the exact system reliability values considered for RRAP with heterogeneous components under the mixed redundancy strategy. Zhang et al. [33] proposed a pseudo parallel genetic algorithm (PPGA) for RAP and RRAP. In Li et al. [34], imperfect nodes are used to formulate global reliability for complex modern network systems. A new

G-mixed strategy is proposed in [35] to solve redundant component issues in reliability optimization problem. An algorithm namely 'BAT-SSOA3' was developed in [36], where a novel RRAP termed as general RRAP was proposed for the series-parallel structure or bridge structure to address several other generalized network structures and for solving the new generalized RRAP. Liu et al. [37] proposed an imperfect switching and a repairman for K-mixed redundancy strategy in 3 components system. Peiravi et al. [38] introduced a new continuous time Markov chain model for both the mixed and k-mixed strategy where lower limit of component can be variable size instead of fixed size.

The main focus of existing RRAP available in the literature is to find the optimal reliability and system redundancy which is relatively blurry. To address the issue, Jianchun et al. [39] introduced a strength-based RAP (SRAP) namely load strength interference model for multi-state system which further utilized a modified ABC algorithm. Zaretalab et al. [40] presented MO-RAP model with choice of selecting suppliers. To optimize the model, they have used the well-known NSGA-II and NRGA Algorithm. Jianchun et al. [41] optimized system reliability for newly introduced reliability based on multi-objective strength RAP using ABC algorithm. Shukla et al. [42] reported a brief study of MFEA for RRAP in an uncertain environment. One might argue that EAs for multi-objective optimization such as Multi-Objective EA (MOEA) are suitable for optimization problem [43] – [45], [46]. However, there are fundamental differences between the principles of MOEA and MFEA paradigm [1]-[3]. The aim of MFEA is to solve multiple tasks simultaneously through implicit parallelism of population and the tasks may or may not have similarity among them.

III. Problem Formulation

The general optimization problem for the reliability redundancy allocation may be mathematically expressed as follows [6], [47]:

$$f = R_S(r, n) \tag{1}$$

$$Subject\ to\ \ g_j(r, n) \leq b, \qquad j = 1, 2, 3, \ldots, k$$

where, $R_S(r, n)$ denotes the total system reliability. Here, r and n defines reliability of the sub-system and number of redundant components for the sub-system, respectively. The j^{th} resource constraint function representing weight, cost and volume of system is denoted by $g_j(r,n)$, where k defines the number of constraints. The reliability function as explained in Eq. (1) changes with respect to the changes in the structure or configuration of the system. For instance, if a system configuration consists m number of sub-systems then the reliability calculation is performed as follows:

$$R_S(r,n) = \prod_{i=1}^{m} \; [1-(1-r_i)^{n_i}] \tag{2}$$

In Eq. (2), i denotes the i^{th} component of the system, r_i represents the reliability of i^{th} component and n_i represents the number of redundant components for i^{th} sub-system. We have considered four RRAP problems under two different test sets. First test set include case studies such as complex (bridge) system, and series-parallel system [48], while the second test set has over-speed system, and life support system in a space capsule. Individually, all the test cases are sequentially labeled as case study 1 - 4. Now, we provide the mathematical description of these four cases.

Case study 1: Complex (bridge) System

The system structure of a complex (bridge) system is depicted in Fig. 2 (a). We have considered five sub-system for this case, which are connected together. The reliability function of this system is non-linear. The mathematical formulation of this system can be written as follows:

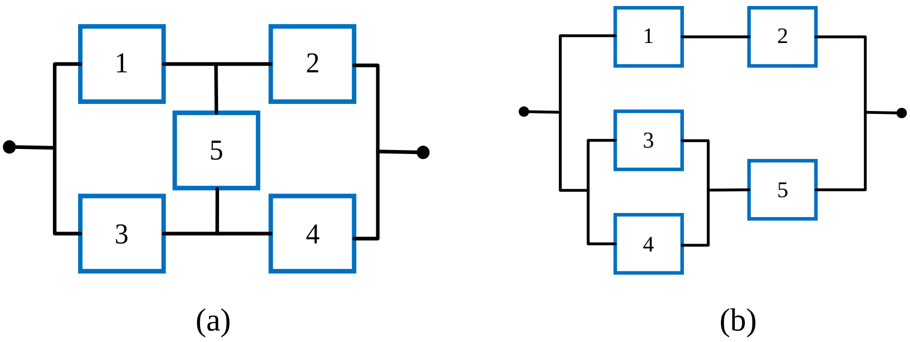


Fig. 2: Schematic diagram of (a) complex (bridge) system and (b) series-parallel system

$$\begin{aligned} Maximize\ f(r,n) = {} & R_1R_2 + R_3R_4 + R_1R_4R_5 + R_2R_3R_5 - R_1R_2R_3R_4 - R_1R_2R_3R_5 \\ & -R_1R_2R_4R_5 - R_1R_3R_4R_5 - R_2R_3R_4R_5 + 2R_1R_2R_3R_4R_5 \end{aligned} \tag{3}$$

$$Subject\ to \qquad g1(r,n) = \sum_{i=1}^{m} w_i v_i^2 n_i^2 - V \leq 0$$

$$g2(r,n) = \sum_{i=1}^{m} \propto_i \left(-\frac{1000}{ln\,(r_i)}\right)^{\beta_i} [n_i + exp\,(0.25n_i)] \; - C \leq 0$$

$$g3(r,n) = \sum_{i=1}^{m} w_i n_i exp\,(0.25 n_i) - W \leq 0$$

$$0 \leq r_i \leq 1, n_i \in Z^+, 1 \leq i \leq m$$

Case study 2: Series-parallel system

The system structure of a series-parallel system is depicted in Fig. 2 (b). Similar to the case study-1, here also five sub-systems are considered and the reliability function is non-linear. The mathematical formulation for this case can be written as follows:

$$Maximize\; f(r,n) = 1 - (1 - R_1 * R_2)[1 - (R_3 + R_4 - R_3 * R_4) * R_5] \qquad (4)$$

$$Subject\ to, \qquad g1(r,n) = \sum_{i=1}^{m} w_i v_i^2 n_i^2 - V \leq 0$$

$$g2(r,n) = \sum_{i=1}^{m} \propto_i \left(-\frac{1000}{ln\,(r_i)}\right)^{\beta_i} [n_i + exp\,(0.25 n_i)] - C \leq 0$$

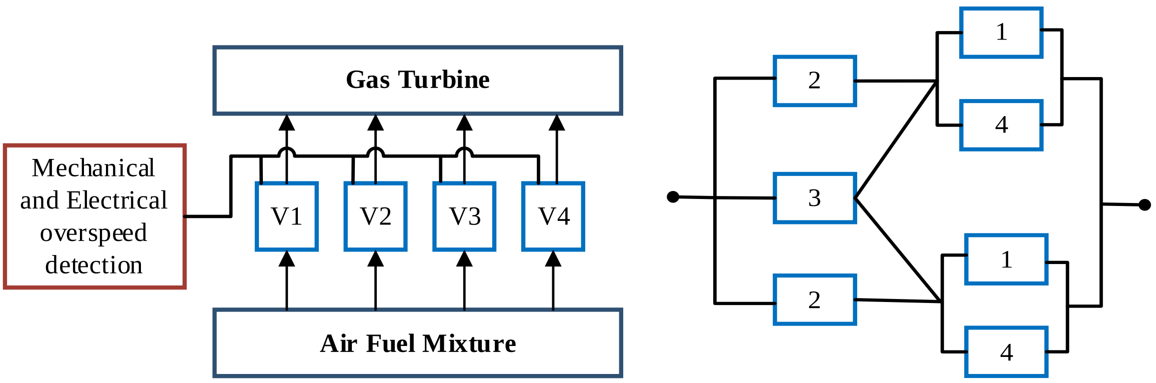
$$g3(r,n) = \sum_{i=1}^{m} w_i n_i exp\,(0.25 n_i) - W \leq 0$$

$$0 \leq r_i \leq 1, n_i \in Z^+, 1 \leq i \leq m$$

In both case study-1 and case study-2, $f(r,n)$ is the overall system reliability.

Case Study 3: Over-speed protection system

This case study considers a gas turbine's over speed protection system (OPS) [49] under the RRAP



(a) (b)

Fig. 3: Schematic diagram of (a) over speed gas turbine system and (b) space capsule.

framework. The fuel supply must be cut-off if the over speed is detected in the system wherein the four valves (V1-V4) need to be shutdown. This control system is modeled as four stage series system. As shown in the Fig. 3 (a), the over-speed protection system is modeled as four stage series system. In such cases, the objective is to find the optimal level of r_i and n_i at each stage of i to achieve the maximum reliability. The mathematical formulation of OPS can be written as follows:

$$Maximize\ R_S(r,n) = \prod_{i=1}^{m} [1-(1-r_i)^{n_i}] \quad (5)$$

$$Subject\ to\ \ g1(r,n) = \sum_{i=1}^{m} w_i v_i^2 n_i^2 - V \le 0$$

$$g2(r,n) = \sum_{i=1}^{m} \propto_i (-\frac{1000}{ln\ (r_i)})^{\beta_i}[n_i + exp\ (0.25n_i)] - C \le 0$$

$$g3(r,n) = \sum_{i=1}^{m} w_i n_i exp\ (0.25n_i) - W \le 0$$

$$0.5 \le r_i \le 1,\ n_i \in Z^+,\ 1 \le i \le m$$

In the above case studies:1-3, m, $r = (r_1, r_2, r_3, \dots, r_m)$ and $n = (n_1, n_2, n_3, .., n_m)$ represents the total number of sub-systems inside the system, reliability and number of components of the systems respectively. The reliability of sub-system i is denoted by R_i. For each sub-system i, its volume, cost and weight are represented by v_i, c_i and w_i, respectively. The upper bound of the sum of the sub-systems volume, cost, and weight are denoted by V, C, and W, respectively. The shaping factor and scaling factor is denoted by $\propto_i$ and β_i. The system volume, system cost and system weight is represented by $g1(r,n)$, $g2(r,n)$ and $g3(r,n)$, respectively.

Case Study 4: Life support system in a space capsule [48]

The life support system in a space capsule is a continuous non-linear optimization problem. The schematic diagram of this problem is shown in Fig. 3 (b). This type of the complex system is mostly found in communication system and the high-pressure oxygen support system in a space capsule [50]. The aim of this problem is to minimize the cost with some specified system reliability. The life support system in a space capsule is consist of four components i.e., $i = 1,2,3,4$ and each component having reliability r_i. Hence, the system reliability can be defined as:

$$R_s = 1 - [(1-r_1)(1-r_4)]^2 - (1-r_3)\big[1 - r_2\big(1-(1-r_1)(1-r_4)\big)\big]^2$$

The mathematical formulation for the life support system in a space capsule can be written as follows:

$$Minimize\ C_s = 2K_1 r_1^{a_1} + 2K_2 r_2^{a_2} + K_3 r_3^{a_3} + 2K_4 r_4^{a_4} \quad (6)$$

$$Subject\ to\ \ R_{s,min} \le R_s \le 1$$

$$R_{i,min} \le r_i \le 1; i = 1,2,3,4$$

In this formulation, the lower bound of the system reliability is denoted as $R_{s,min}$ with value 0.9. Similarly, the lower bound of i^{th} component's reliability is denoted as $R_{i,min}$ and value assigned is 0.5.

IV. Solution Approach

EAs being the population based meta-heuristic algorithms are generally designed for solving two types of problems: single objective problem (SOP) [42] and multi-objective problem (MOP) [43], [44], [51]. A recent EA based framework, named MFEA, provides an architecture for solving multiple optimization tasks simultaneously [1]. It has been successfully applied to many real-world problems (for example, [52]).

Multi-Factorial Evolutionary Algorithm (MFEA)

This sub-section provides the detailed explanation of the MFEA. The two key purposes of solving multiple tasks at the same time are: i) a unified search space representation based on task's dimension, and ii) a unique optimization function using a common set of population. In a multi-tasking environment, solving a simple optimization task may complement more complex task through computationally encoded knowledge (or genetic materials) transfer. Since, MFEA utilize the unified search space, it helps to transfer knowledge between tasks to acquire the optimal solution, which also lead to reduction in memory consumption and faster convergence rate [1].

Let's assume K be the number of tasks, where T_j defines the j^{th} task and each task has its own search space, denoted as X_j. The objective function of T_j will be $f_j \rightarrow R$. In MFEA, the multi culture environment is maintained by Multi Factorial Optimization (MFO). The formulation of MFO can be written as:

$$\{x_1, x_2, \ldots x_{k-1}, x_k\} = arg\ min\, \{f_1(x), f_2(x) \ldots\ldots f_{k-1}(x), f_k(x)\}$$

where, x_j is the feasible solution of X_j and f_j is treated as an additional factor which influences the evolution of a single population of individuals. The combination of all feasible solutions become a single search space, termed as, unified search space [1]. The j^{th} task dimension is defined as d_j. The dimension of unified search space is D. Although different task has different dimension, D will take the maximum dimension among the tasks, i.e., $D = max\,\{d_1, d_2 \dots d_k\}$. The unified search space with dimension D has the normalized value range from 0 to 1.

Algorithm 1 provides the basic working structure of the MFEA algorithm. The algorithm starts with generating the initial population. Each individual has some common properties for task specific objective evaluation. Here is some basic definition of individual population properties:

i) ***Factorial Cost (*** Ψ_j^i ***)*** is an objective function value (f_j) of task (T_j) for a population individual P_i.

ii) ***Factorial Rank*** (r_j^i) depicts the rank of population individual P_i on task (T_j) based on the sorted objective function value with respect to (Ψ_j).

iii) ***Scalar Fitness***(ϕ_i) indicates the individual rank over all tasks. Scalar fitness of individual pi is the inverse of $rj\ i\ gven\ by\ \phi_i = \frac{1}{r_j^i}$.

iv) **Skill F*actor*** (τ) is the index of task t_j for the individual P_i, which indicates the individual is best suited for corresponding tasks only. After generating the population, initially each individual is evaluated for every task to decide its task group based on the corresponding skill factor. As each individual only belongs to one specific task, it helps to generate new solutions/offspring in an effective manner.

Algorithm 1: *MFEA procedure*

---

i. Generate initial population randomly and set as current-population (P)
ii. Population evaluation based on all the task in multi task environment.
iii. Skill-factor (τ) *computation of population-individual for dividing population into task specific group.*
While *(not satisfied stopping condition?)*
  a. To generate offspring-population(C) *apply genetic operations on P. Refer to Algorithm 2.*
  b. Evolution of C with respect to specific task only. Refer to Algorithm 3
  c. Form the intermediate-population (U) using the combination of P and C.
  d. Update both scalar fitness (ϕ) *and τ of for every individual of U.*
  e. Select best individuals from U based on the ϕ *to form next P.*
End while

Algorithm 2 *Assortative Mating*

---

i. To generate offspring P_a *and* P_b *randomly selected from* P *as parents.*
ii. A random number (n) *is generated within the range 0-1.*
if *(*$\tau_a == \tau_b$*) or (n < rmp)* ***then***
  Crossover between P_a *and* P_b *will generate* C_a *and* C_a.
Else
  A slightly Mutation on P_a *and* P_b *will generate* C_a *and* C_b *independently.*
End if

Algorithm 3 *Vertical cultural transmission*

i. *An offspring c will either generates from crossover of* P_a *and* P_b *it will generate from mutation of any* P_a *or* P_b.

if *(c belong to both* P_a *and* P_b *)*

a. *A random number* (n) *is generated within the range 0-1.*

if $(n < 0.5)$ *then*

τ_a *Will be assigned as skill-factor of c.*

Else

τ_b *Will be assigned as skill-factor of c.*

End if

Else *(c belongs to any of the* P_a *and* P_b *)*

b. *Skill-factor of c will be its parent skill factor.*

End if

In *Algorithm 2*, the *assortative-mating* is implemented for generating a new population, that implies parents having similar skill-factor or some satisfactory level of random mating probability (RMP), which will allow them for crossover else they will generate a new population through mutation. In *Algorithm 3, vertical cultural transmission* has been implemented, which is the process through which the offspring imitates their parent's skill factor. This cultural transmission ensures task specific evaluation of offspring throughout the generation. The combination of offspring and current population will form an intermediate population. Based on the scalar fitness (ϕ), the population for next generations will be selected from the intermediate population, which is known as the best population of that generation. The best individual of the final generation is considered as an optimal solution.

Elements of the proposed MFEA

This sub-section discusses the proposed MFEA-based solution approach to solve two tasks. Here we have considered test set-1, which includes two different optimization problems taken together, viz., Case Study-1 and Case Study-2. These two optimization problems are treated as two different tasks of MFEA. In a multi-tasking environment, solving two tasks simultaneously allows the genetic materials between tasks to complement each other. It is based on the similarity between tasks (i.e., similar system structure, similar decision variable), which will ultimately lead to faster convergence of task and reduces the consumption of memory [1].

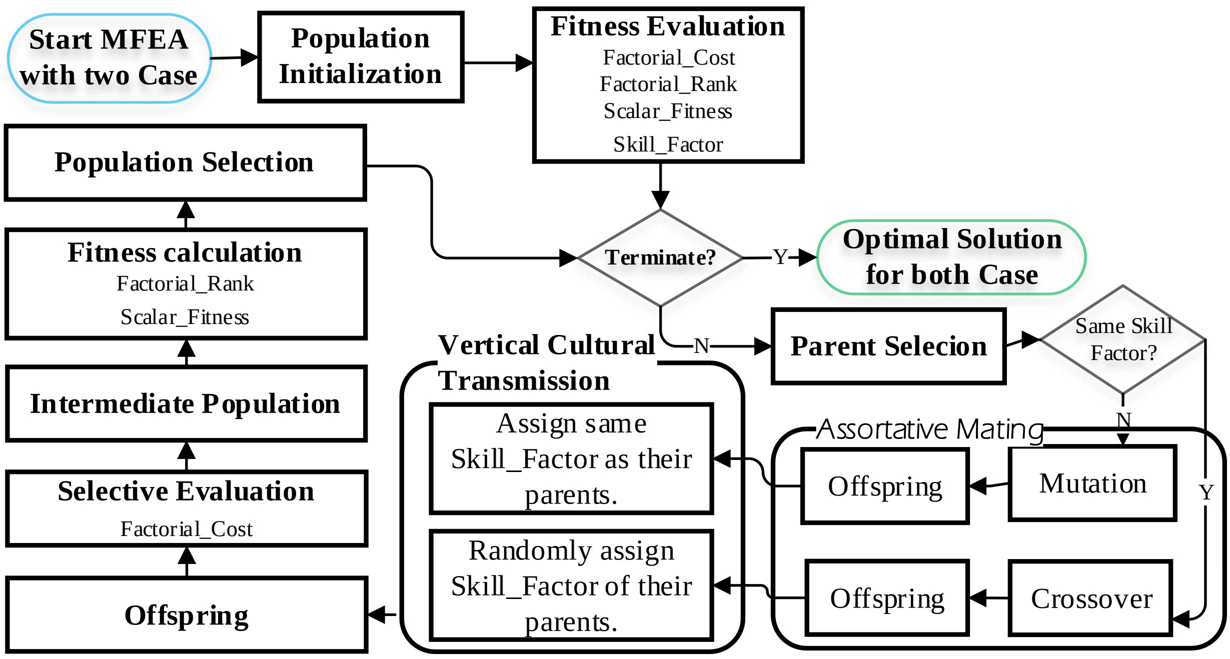


Fig. 4: Working of MFEA for RRAP.

Similar to other EAs, MFEA also starts with the encoding of real-valued chromosomes, followed by a set of population initialization. Each individual in the population is evaluated for both the tasks to determine its skill factor through factorial-rank which ultimately helps to calculate the fitness value for the individual**.** Further, the elite/best solution individual is selected from the current population as parent to generate the new population offspring through genetic operations, namely, crossover and mutation. This optimization process continues to evolve until the given termination condition is satisfied. Fig. 4 shows the flowchart to visualize the working of our proposed MFEA based solution approach for RRAP. We will now discuss in detail about the major steps associated with designing the MFEA framework for solving RRAP cases.

Stage 1: Population initialization:

The dimension of a chromosome is the maximum dimension of the task. For instance, in case of first test set, both the case studies (or optimization tasks) have similar numbers of decision variables, i.e., five decision variables for component reliability and five decision variables for number of redundant components. We have interchangeably used case studies and tasks throughout the manuscript. Here,

0.75	**0.90**	**0.86**	**0.73**	**0.65**	**3**	**2**	**3**	**3**	**2**

Component Reliability **Number of Component**

Fig. 5: A chromosome model for MFEA encoding.

Prior to the population initialization, the real valued genetic encoding of the chromosome is performed. the dimension of the chromosome will be ten. An example of a chromosome encoding for the addressed problem is illustrated in Fig. 5. Each chromosome is linked to the properties of factorial rank, factorial cost, scalar fitness and skill factor. Task specific population conversion is performed to optimize both the case studies. In our approach, we have utilized the decoding scheme of continuous problems for the population conversion [1].

Stage 2: Fitness evaluation and new population selection:

At first, for both the tasks, the factorial cost is evaluated by the task specific objective function (Eqs. (3) & (4)). Secondly, the ascending order sorting of factorial cost with respect to both the tasks gives task specific factorial rank for every chromosome. And finally, fitness value of a chromosome, namely, scalar-fitness value is computed on the basis of its best rank over both the tasks. The assignment of the skill factor to each chromosome is performed based on the factorial rank to determine the best fit solution for the task. Finally, the chromosome's fitness evaluation is performed on the basis of a computed skill factor. The new population selection for the current generation is based on the elitist selection method, which ensures the survival of the best population during every iteration.

Stage 3: Generating new population:

Two individuals are selected randomly as parents for generating new offspring. The selected parents go through the most important genetic process namely *assortative mating (Algorithm 2)* and *vertical cultural transmission (Algorithm 3)*. The assortative mating helps to mate with different task groups which allow implicit knowledge transfer between both the tasks. New population is generated using crossover if the parents have the same skill factor or the given random mating probability (RMP) is satisfied. Otherwise, the new population will be generated through mutation. The vertical cultural transmission specifies the newly generated offspring's identity by assigning its skill factor the same as their parents. If the new population is generated by crossover, then randomly either of the parent's skill factors will assign to the new child. Otherwise, if the new population is generated through mutation, then the assignment of new offspring's skill factor will be similar to their parent's skill factor. The generated solution and existing population are truncated into new population for the next iteration.

Stage 4: Termination Criteria:

The procedure will terminate after a maximum number of generations is reached. Otherwise, the procedure will repeat from Stage 2.

V. EXPERIMENTS, RESULT AND COMPARATIVE ANALYSIS

For the simulation, we have considered two test sets of RRAP problems, which are solved using the proposed approach. Each test set includes two numerical problems of RRAP and their respective benchmark dataset, which are discussed in Section III. As discussed above, the two optimization problems in both test sets are considered as two different tasks of MFEA. We have also solved each case studies within test sets separately using GA [53], PSO [54], SA [55], DE [56] and ACO [57] for comparative analysis. The hardware specification for the execution consists of a computer having 8GB RAM and an Intel(R) Core (TM) i5-2400 CPU with a licensed MATLAB 2018b. Each algorithm has been executed for 20 independent runs.

(a) Parameter setting:

The size of the initial population is considered as 50 and the evolution for maximum generation is set as 1000 for every algorithm. The crossover probability (RMP) is set as 0.8 [1], [58], [59]. For the proposed MFEA approach, the selection is based on an elitist selection method which indicates that a better fitness valued population individual will be selected as parent; whereas in case of GA, PSO and such others, the parent selection method for generating offspring is based on tournament selection [47].

(b) Input Data:

Table I and Table II contains five separate input sets of data used in experiments for case study-1 and case study-2, respectively. It also provides the values for shaping factor, scaling factor, weight component, volume component, volume upper bound, cost bound, and weight bound for all 5 different sub-systems. Similarly, Table III contains four separate input sets of data and provides values of different parameters used for experiments for case study-3. The value of K_1, K_2, K_3, and K_4 is set as 100, 100, 200, 150, respectively for case study-4. In addition, the values for a_i are set as 0.6. The obtained values from the experiments of both the test sets are noted for each independent run of MFEA, PSO, GA, SA, DE and ACO.

Table I. Input data for case study-1

i	$10^5 x_i$	β_i	w_i	$w_i v_i^2$	V	C	W
1	2.330	1.5	7	1			
2	1.450	1.5	8	2			
3	0.541	1.5	8	3	110	175	200
4	8.050	1.5	6	4			
5	1.950	1.5	9	2			

Table II. Input data for case study-2

i	$10^5 x_i$	β_i	w_i	$w_i v_i^2$	V	C	W
1	2.500	1.5	2	3.5			
2	1.450	1.5	4	4			
3	0.541	1.5	5	4	180	175	100
4	0.541	1.5	8	3.5			
5	2.100	1.5	4	3.5			

Table III. Input data for case study-3

i	$10^5 x_i$	β_i	w_i	$w_i v_i^2$	V	C	W
1	1.0	1.5	6	1			
2	2.3	1.5	6	2			
3	0.3	1.5	8	3	250	400	500
4	2.3	1.5	7	2			

(c) Results of test set- 1 (case study-1 and case study-2):

Table IV and Table V provides the comparative analysis of the average and best reliability values generated by MFEA, GA, PSO, SA, DE and ACO for case study -1 and case study-2, respectively. The results at the interval of 100 generations are shown in these tables for illustrating the progress more clearly towards convergence. Table VI compiles the best solutions achieved from the executions of MFEA, PSO, GA, SA, DE and ACO for the both cases of first test set. It also provides several measurement values associated with the best solution namely components' reliability, number of redundant components, complete system reliability, along with constraints values such as volume, weight and cost. The constraints values obtained by the MFEA and PSO are better than the GA, SA, DE and ACO in the case study-1. In case study-2, MFEA and the rest of all algorithms are providing similar constraints value along with better reliability, weight and volume values.

Table IV. Average and best reliability values over varying iterations for the case study-1.

Algorithm	Convergence	Iterations									
		100	**200**	**300**	**400**	**500**	**600**	**700**	**800**	**900**	**1000**
MFEA	Average	0.997099	0.998277	0.998779	0.999070	0.999604	0.999675	0.999693	0.999694	0.999694	0.999695
	Best	0.996520	0.999134	0.999755	0.999850	0.999851	0.999851	0.999851	0.999851	0.999851	0.999851
PSO	Average	0.949824	0.949825	0.949825	0.999814	0.999817	0.999818	0.999818	0.999818	0.999818	0.999818
	Best	0.999865	0.999865	0.999865	0.999865	0.999865	0.999865	0.999865	0.999865	0.999865	0.999865
GA	Average	0.999644	0.999644	0.999644	0.999644	0.999644	0.999644	0.999644	0.999644	0.999644	0.999644
	Best	0.999835	0.999835	0.999835	0.999835	0.999835	0.999835	0.999835	0.999835	0.999835	0.999835
SA	Average	0.999869	0.999870	0.999870	0.999870	0.999870	0.999870	0.999870	0.999870	0.999870	0.999870
	Best	0.999889	0.999890	0.999890	0.999890	0.999890	0.999890	0.999890	0.999890	0.999890	0.999890
DE	Average	0.999473	0.999817	0.999858	0.999872	0.999881	0.999883	0.999884	0.999885	0.999887	0.999887
	Best	0.999580	0.999847	0.999872	0.999879	0.999887	0.999887	0.999887	0.999887	0.999889	0.999889
ACO	Average	0.999798	0.999877	0.999884	0.999887	0.999888	0.999889	0.999889	0.999889	0.999889	0.999890
	Best	0.999810	0.999865	0.999888	0.999889	0.999889	0.999890	0.999890	0.999890	0.999890	0.999890

Table V. Average and best reliability values over varying iterations for the case study-2.

Algorithm	Convergence	Iterations									
		100	**200**	**300**	**400**	**500**	**600**	**700**	**800**	**900**	**1000**
MFEA	Average	0.999706	0.999843	0.999889	0.999923	0.999927	0.999928	0.999928	0.999928	0.999928	0.999851
	Best	0.999968	0.999980	0.999984	0.999984	0.999984	0.999984	0.999984	0.999984	0.999984	0.999984
PSO	Average	0.949969	0.949969	0.999960	0.999966	0.999967	0.999967	0.999967	0.999967	0.999967	0.999967
	Best	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986
GA	Average	0.996027	0.996028	0.996031	0.996035	0.996035	0.996035	0.996035	0.996035	0.996036	0.996036
	Best	0.999984	0.999984	0.999984	0.999984	0.999984	0.999984	0.999984	0.999984	0.999984	0.999984
SA	Average	0.999982	0.999982	0.999982	0.999982	0.999982	0.999982	0.999982	0.999982	0.999982	0.999982
	Best	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986
DE	Average	0.999923	0.999976	0.999981	0.999983	0.999984	0.999985	0.999985	0.999986	0.999986	0.999986
	Best	0.999911	0.999984	0.999984	0.999985	0.999985	0.999985	0.999986	0.999986	0.999986	0.999986
ACO	Average	0.999968	0.999983	0.999984	0.999984	0.999984	0.999985	0.999985	0.999985	0.999985	0.999985
	Best	0.999978	0.999983	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986	0.999986

Figs. 6 (a) & (b) shows the plots for the average reliability values obtained using MFEA, PSO, GA, SA, DE and ACO for case study-1 and case study-2, respectively. It can be observed from Fig. 6(a) that MFEA is able to produce better solutions from the beginning of the execution and led to convergence within 500 generations. On the other hand, PSO converges after 400 generations with the slow progress in the evolution process. However, all algorithms are performing better than PSO in generating better solutions from the earlier stage. Fig. 6 (b) shows the average reliability values obtained from MFEA,

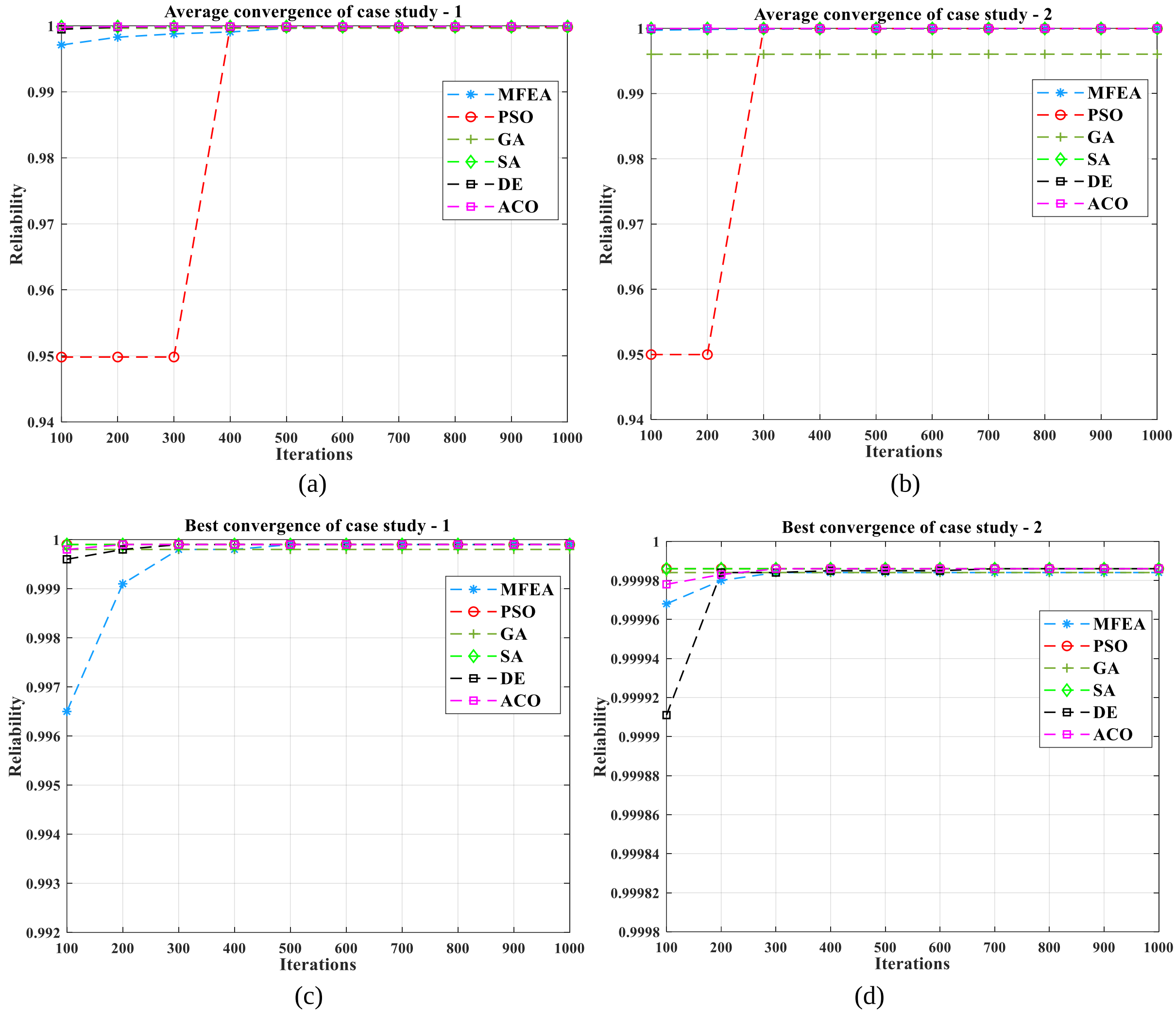


Fig. 6: MFEA vs. PSO vs. GA vs SA vs DE vs ACO comparison a) Average reliability values for case study-1; b) Average reliability values for case study-2; c) Best reliability values for case study-1, & (d) Best reliability values for case study-2

Table VI. Comparison of the best solutions obtained for case study-1 and case study-2

Case study	Algorithm	Component reliability (r)	Redundant component	System Reliability (R)	Volume	Weight	Cost
1	**MFEA**	0.8256, 0.8636, 0.8935, 0.7001, 0.7699	2, 3, 3, 3, 2	0.999851	93.00	192.4811	175.00
	PSO	0.8232, 0.8744, 0.9102, 0.6749, 0.6701	3, 3, 2, 3, 2	0.999865	83.00	189.427	175.00
	GA	0.8063, 0.8936, 0.8305, 0.7212, 0.7477	3, 3, 3, 2, 2	0.999835	78.00	195.53	175.00
	SA	0.8280, 0.8578, 0.9142, 0.6481, 0.7049	3, 3, 2, 4, 1	0.999890	105.00	198.44	175.00
	DE	0.8284, 0.8630, 0.9115, 0.6440, 0.6986	3, 3, 2, 4, 1	0.999889	105.00	198.44	174.92
	ACO	0.8288, 0.8571, 0.9138, 0.6491, 0.7021	3, 3, 2, 4, 1	0.999890	105.00	198.44	175.00
2	**MFEA**	0.8166, 0.8424, 0.8582, 0.8635, 0.8770	2, 2, 2, 3, 4	0.999984	180.00	98.21	175.00
	PSO	0.7764, 0.8730, 0.8902, 0.8913, 0.8631	3, 2, 2, 2, 4	0.999986	150.00	98.21	175.00
	GA	0.7332, 0.8192, 0.9119, 0.8854, 0.8786	3, 2, 2, 2, 4	0.999984	150.00	98.21	175.00
	SA	0.7753, 0.8713, 0.8912, 0.8912, 0.8630	3, 2, 2, 2, 4	0.999986	150.00	98.21	175.00
	DE	0.7791, 0.8687, 0.8837, 0.8923, 0.8644	3, 2, 2, 2, 4	0.999986	150.00	98.21	174.96
	ACO	0.7755, 0.8717, 0.8910, 0.8916, 0.8627	3, 2, 2, 2, 4	0.999986	150.00	98.21	175.00

PSO, GA, SA, DE and ACO for case study-2. MFEA, GA, SA, DE and ACO are able to generate better solutions from the very beginning of the evolution process and converge within 100 generations. In the case of PSO, it converges at 500 generations with better reliability value like other algorithms. Similarly, Figs. 6 (c) & (d) illustrates the best convergence plots of reliability values with varying iterations of all six algorithms for case study-1 and case study-2, respectively. From both the figures, it is evident that MFEA generates solutions with optimal reliability value similar to all other algorithms. From the analysis and discussion, we can infer that the proposed MFEA framework is able to solve two RRAP problems simultaneously compared to solving each case study separately.

(d) Results of test set- 2 (case study-3 and case study-3):

Similar to first test set, Tables VII & VIII provides the comparative analysis for the average and best reliability values at the interval of 100 generations for the second test set. Corresponding to these tables, Figs. 7(a) & (c) shows the graph plot for average and best convergence of reliability values for case study-3, while Figs. 7(b) & (d) depicts the respective figures for case study-4. Table IX gives insight about the best solution generated using each of the six algorithms for the case study 3 & 4. It also provides decision variables associated with the best solution. It can be seen that solutions generated

Table VII. Average and best reliability values over varying iterations for the case study-3.

Algorithm	**Convergence**	**Iterations**									
		100	**200**	**300**	**400**	**500**	**600**	**700**	**800**	**900**	**1000**
MFEA	Average	0.999784	0.999869	0.999883	0.999885	0.999886	0.999896	0.999899	0.999901	0.999901	0.999911
	Best	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955
PSO	Average	0.999945	0.999946	0.999946	0.999946	0.999946	0.999946	0.999946	0.999946	0.999946	0.999946
	Best	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955
GA	Average	0.999888	0.999888	0.999888	0.999888	0.999888	0.999888	0.999888	0.999888	0.999888	0.999888
	Best	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955
SA	Average	0.999875	0.999875	0.999875	0.999875	0.999875	0.999875	0.999875	0.999875	0.999875	0.999875
	Best	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955
DE	Average	0.999909	0.999945	0.999950	0.999952	0.999953	0.999954	0.999954	0.999954	0.999954	0.999955
	Best	0.999915	0.999950	0.999954	0.999954	0.999954	0.999954	0.999954	0.999955	0.999955	0.999955
ACO	Average	0.999942	0.999950	0.999951	0.999951	0.999951	0.999951	0.999951	0.999951	0.999951	0.999951
	Best	0.999945	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955	0.999955

by each of these algorithms are feasible and give almost similar results in case study-3. Similarly, in case study-4, the decision variables generated by all six algorithms are feasible. The system reliability value obtained from the decision variables value shows that MFEA, PSO, SE and DE give similar but comparatively better results than GA and ACO. Furthermore, MFEA, SA, and ACO produce marginally better cost value in comparison to GA, PSO and DE.

Table VIII. Average and best reliability values over varying iterations for case study-4

Algorithm	Convergence	Iterations									
		100	200	300	400	500	600	700	800	900	1000
MFEA	Average	0.912529	0.914954	0.963362	0.963745	0.963818	0.963846	0.963875	0.963880	0.963897	0.963899
	Best	0.962848	0.963696	0.963843	0.963871	0.963901	0.963901	0.963905	0.963905	0.963905	0.963905
PSO	Average	0.963461	0.963570	0.963605	0.963664	0.963682	0.963689	0.963696	0.963704	0.963710	0.963725
	Best	0.963663	0.963887	0.963905	0.963905	0.963905	0.963905	0.963905	0.963905	0.963905	0.963905
GA	Average	0.955269	0.955272	0.955272	0.955272	0.955272	0.955272	0.955272	0.955272	0.955272	0.955272
	Best	0.963889	0.963889	0.963889	0.963889	0.963889	0.963889	0.963889	0.963889	0.963889	0.963889
SA	Average	0.963901	0.963901	0.963901	0.963901	0.963901	0.963901	0.963901	0.963901	0.963901	0.963901
	Best	0.963905	0.963905	0.963905	0.963905	0.963905	0.963905	0.963905	0.963905	0.963905	0.963905
DE	Average	0.962728	0.963659	0.963821	0.963860	0.963876	0.963885	0.963891	0.963891	0.963895	0.963896
	Best	0.962775	0.963851	0.963877	0.963877	0.963883	0.963889	0.963903	0.963903	0.963903	0.963905
ACO	Average	0.963893	0.963893	0.963894	0.963894	0.963894	0.963894	0.963894	0.963894	0.963894	0.963894
	Best	0.963901	0.963901	0.963902	0.963902	0.963903	0.963903	0.963903	0.963904	0.963904	0.963904

Table IX. MFEA vs. PSO vs. GA vs SA vs DE vs ACO: analyzing the best solutions of case studies 3 & 4

Case study	Algorithm	Component reliability (r)	Redundant component	System Reliability (R)	Volume	Weight	Cost
3	**MFEA**	0.9022, 0.8870, 0.9485, 0.8509	5, 5, 4, 6	0.999955	195.00	484.64	400.00
	PSO	0.9017, 0.8499, 0.9481, 0.8882	5, 6, 4, 5	0.999955	195.00	475.20	400.00
	GA	0.8999, 0.8490, 0.9482, 0.8897	4, 5, 3, 4	0.999955	195.00	475.20	400.00
	SA	0.9016, 0.8882, 0.9481, 0.8499	5, 5, 4, 6	0.999955	195.00	484.64	400.00
	DE	0.9014, 0.8483, 0.9490, 0.8886	5, 6, 4, 5	0.999955	195.00	475.20	399.94
	ACO	0.9016, 0.8498, 0.9481, 0.8882	5, 6, 4, 5	0.999955	195.00	475.20	399.99
4	**MFEA**	0.6272, 0.9912, 0.5000, 0.5000	-	0.963905	-	-	679.99
	PSO	0.6258, 0.9929, 0.5000, 0.5000	-	0.963905	-	-	680.00
	GA	0.6337, 0.9834, 0.5000, 0.5000	-	0.963889	-	-	680.00
	SA	0.6276, 0.9906, 0.5000, 0.5000	-	0.963905	-	-	679.99
	DE	0.6267, 0.9918, 0.5000, 0.5000	-	0.963905	-	-	680.00
	ACO	0.6250, 0.9939, 0.5000, 0.5000	-	0.963904	-	-	679.99

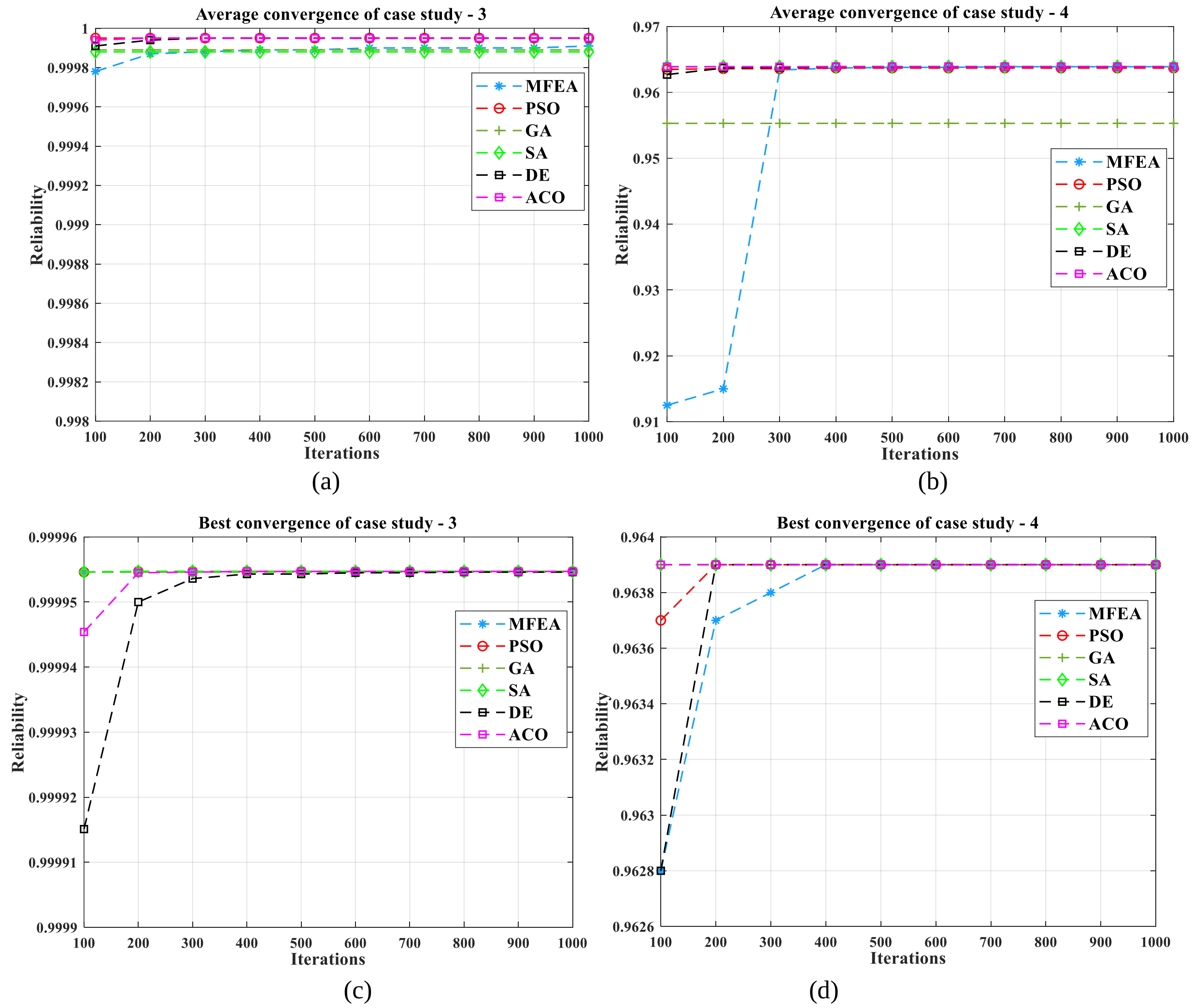


Fig. 7: MFEA vs. PSO vs. GA vs SA vs DE vs ACO comparison a) Average reliability values for case study-3; b) Average reliability values for case study-4; c) Best reliability values for case study-3, & (d) Best reliability values for case study-4

Fig. 7(a) reflects the faster convergence of MFEA and DE from 200 generations. The PSO, GA, SA and ACO are able to converge at the start of the iteration. All six evolutionary approaches produce almost similar reliability values. Fig. 7 (b) shows the plot of average reliability values obtained for case study-4. Here, MFEA converged at 400 generations and produced better solutions. The PSO, GA, SA and ACO converge before 100 generations while DE converges at 200 generations. Although MFEA took more generations to converge in comparison to other algorithms, it has generated better solutions than other algorithms in addition to the fact that it has solved two problems concurrently.

In Fig-7 (c), the convergence plot of best reliability values obtained by six evolutionary algorithms for case study-3 are shown. The MFEA, PSO, GA and SA are able to generate good solutions and converge within 100 generations. DE and ACO converged at 800 and 200 generations, respectively.

Similarly, Fig. 7(d) shows the best convergence plot of reliability values for case study-4. Here, MFEA has generated comparable solutions to PSO, SA, DE and ACO, while it converged at 500 generations. Only GA converged faster, within 100 generations.

(e) Comparing the computation times and the ranking of the algorithms:

We have now provided the computation time analysis of MFEA and its comparison with the other approaches such as: PSO, GA, SA, DE, and ACO. Moreover, we have also ranked these algorithms using TOPSIS approach [14] which is evaluated on the basis of computation time, average and best reliability values obtained using every algorithm.

(i) Comparing the computation times:

Table X gives the comparative computation time values while solving each test set using MFEA, PSO, GA, SA, DE and ACO. It also provides the percentage improvements in computation time of proposed MFEA based approach as compared to all other approaches. It can be noticed that the computation time for MFEA is the least among all other five algorithms for solving each of the test sets, which is 86.74 *sec* and 54.20 *sec*, respectively. The running time of GA and ACO are much nearer to each other for every set and same is the case for the PSO and DE. The evolutionary process of SA is much slower compared to all other approaches, taking the longest computation time among all, i.e., 4446.01 *sec* and 1497 *sec* for first and second test sets, respectively.

Table X. Computation time of MFEA, PSO, GA, SA, DE and ACO for first and second test sets

Approach	Total computation time		Improvements in the computation time of the proposed MFEA based approach	
	Test set 1 (case studies 1 and 2)	Test set 2 (case studies 3 and 4)	Test set 1 (case studies 1 and 2)	Test set 2 (case studies 3 and 4)
Proposed MFEA based approach	**86.74**	**54.20**	-	-
GA based approach	120.51	63.34	28.02%	14.43%
ACO based approach	125.26	63.21	30.75%	14.25%
PSO based approach	149.94	78.83	42.15%	31.24%
DE based approach	150.37	79.12	42.32%	31.50%
SA based approach	4446.1	1497.22	98.05%	96.38%

Also, the quantitative analysis of the proposed MFEA based approach in comparison to GA, ACO, PSO, DE and SA showed significant percentage improvements of 28.02%, 30.75%, 42.15%, 42.32% and 98.05% for first test set, and 14.43%, 14.25%, 31.24%, 31.50% and 96.38% for second test set, respectively. Therefore, MFEA outperforms all other algorithms in terms of computation time, which establishes its better suitability to solve multiple RRAP problems together considering its at par reliability with faster computation.

(ii) Ranking of algorithms based on the multi-criteria decision making using TOPSIS Method:

In this section, we analyze and evaluate all the considered methods for multi-criteria decision making using TOPSIS method [14]. The three performance indicators i.e., computation time, average and best reliability values obtained using every algorithm for test sets is adopted as an input to conduct TOPSIS. The steps of the straight forward TOPSIS approach is mentioned in Yadav et al. [14], where S_i^+ and S_i^- represents the separation measure and P_i represents relative closeness to the ideal solution. These are also termed as performance score which we have computed for our approaches and ranking is evaluated in the decreasing order of P_i. Note that, each of these measurements are equally significant for evaluation, therefore, an equal weightage of 0.33 is assigned to each of them. Tables XI & XII shows the performance score and ranks of the algorithms for first and second task set, respectively. From Tables XI & XII, it can be clearly seen that MFEA holds top rank with best score among all other algorithms for both test sets, following which GA ranks second and PSO is on the third rank. Similarly, the fourth rank is held by DE for both test sets and ACO bearing the 5th rank determines its worst performance among all approaches. Hence, from overall analysis, we can state that MFEA based approach is not only better in solving the problem quicker than other but also performs the best.

Table XI. Performance score and rank of algorithms for first task set

Algorithm	S_i^+	S_i^-	P_i	**Rank**
MFEA	0.00002421488516	0.342455025	0.9999292953	1
PSO	0.004964908577	0.3374901398	0.9855020138	3
GA	0.00268338198	0.3398020396	0.9921649746	2
SA	0.3424548033	0.0004036779454	0.001177389411	6
DE	0.004998255304	0.33745679	0.9854046381	4
ACO	0.3394294283	0.144728214	0.2989278725	5

Table XII. Performance score and rank of algorithms for second task set

Algorithm	S_i^+	S_i^-	P_i	Rank
MFEA	0.000004496091187	0.3355814168	0.99999	1
PSO	0.005727859313	0.329853559	0.9829315361	3
GA	0.002314254666	0.3334545214	0.9931075942	2
SA	0.3355801698	0.0009150667394	0.002719404734	6
DE	0.005795267174	0.3297861707	0.9827306683	4
ACO	0.3334863713	0.1451951766	0.3033231117	5

(f) Statistical significance test:

To observe the statistical differences among the algorithms, we have conducted a one-way ANOVA test using the optimum reliability values found in last generations for each independent run [47]. Fig. 8 (a)-(d) shows the results obtained by ANOVA test for the case studies-1 to 4, respectively.

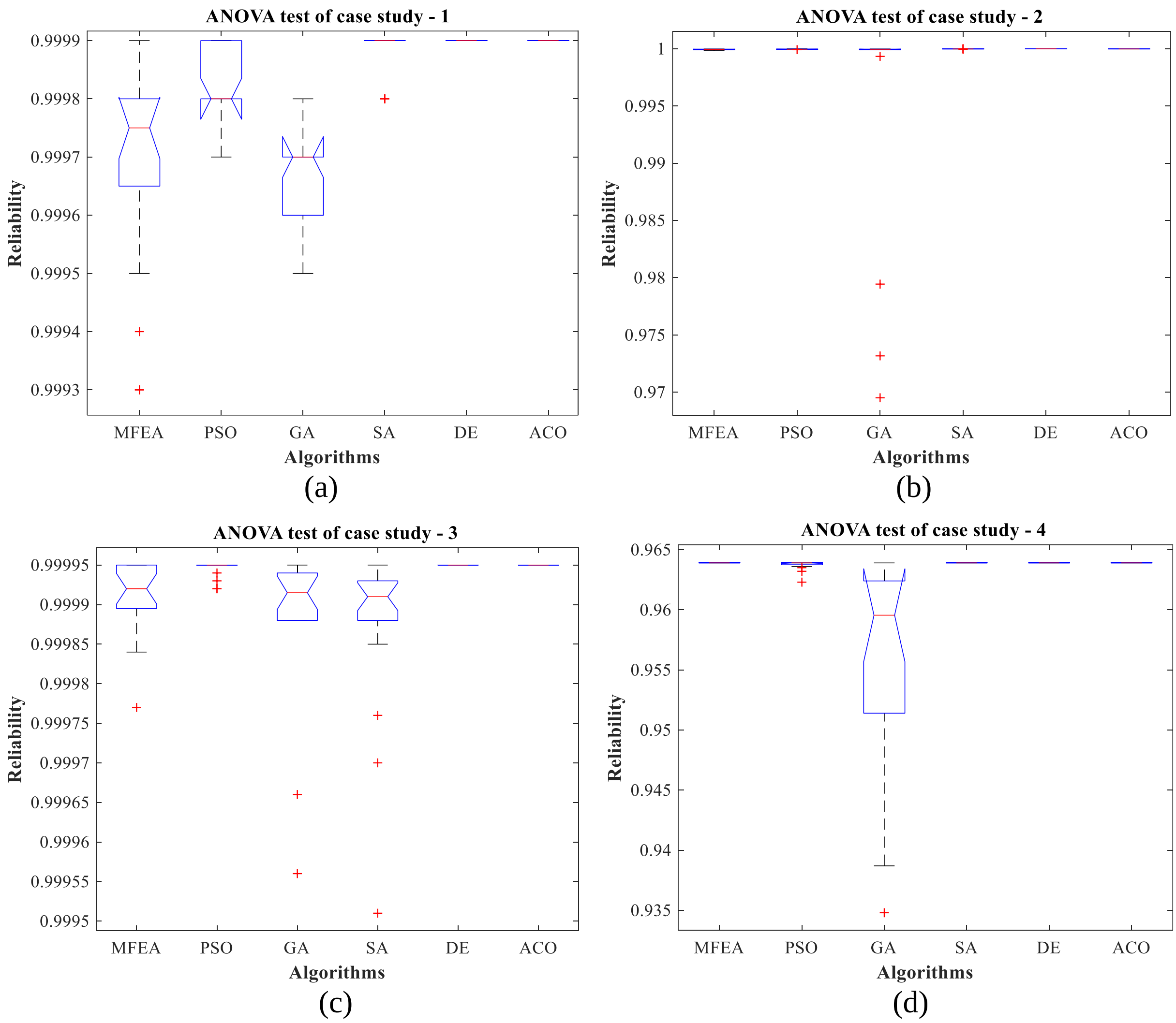


Fig. 8: Comparative results of ANOVA test based on best reliability value obtained in 20 independent runs of MFEA, PSO, GA, SA, DE and ACO in (a) Case study – 1 (b) Case study – 2 (c) Case study – 3 (d) Case study – 4.

In Fig. 8, each figure includes box plots for visualizing the median of respective algorithms reliability value. Fig. 8(a) demonstrates the statistical significance with median values for case study-1 for all the algorithms. It can be observed that MFEA, PSO, SA, DE and ACO have similar median values, whereas the median value obtained using MFEA is better than GA. Similarly, for the case study-2, the median values for all six evolutionary algorithms are almost equal, and statistically performances of all the algorithms (except GA) are similar to MFEA, as shown in Fig. 8(b). The median values of MFEA, GA and SA are almost similar, and the other three algorithms, PSO, DE and ACO, have slightly better median values for the case study-3, as shown in Fig. 8(c). Moreover, GA and SA are statistically different from PSO, DE and ACO. But, our proposed MFEA based approach has no statistical difference with them. For the case study-4, the median values of MFEA, PSO, SA, DE and ACO are almost similar, whereas GA is far away from the rest of the other five algorithms, as shown in Fig. 8(d).

From the overall comparative analysis of results obtained for each case, we can state that MFEA performs better in terms of producing quality solutions, faster convergence and statistical significance. In addition, the MFEA based solution approach has been quite successful in solving two RRAP problems simultaneously. Thus, the study establishes the effective applicability of MFEA to the domain of study to solve multiple RRAP problems together (without any significant compromise on the solution quality) rather than solving each problem individually.

VI. CONCLUSION AND FUTURE WORK

This paper has proposed a novel approach to solve the reliability redundancy allocation problem. The proposed method is based on the framework of single-population based Evolutionary multi-task optimization i.e., multifactorial evolutionary algorithm (MFEA). To observe the applicability and the usefulness of our proposed approach, we have first solved two RRAP cases: series-parallel system and complex bridge system, simultaneously. Then, we have attempted to solve another set of RRAPs, namely, the over-speed protection system and the life support system in a space capsule, simultaneously, in the same solution approach based on the proposed MFEA framework. The well-known single objective evolutionary and nature inspired optimizers PSO, GA, SA, DE, and ACO are also implemented for comparative analysis with the proposed approach. The experimental results illustrate the effectiveness of the proposed MFEA framework-based approach where the generated solutions are feasible and competitive for applicability in the domain of RRAP. In addition, the proposed MFEA approach is able to effectively solve multiple cases of RRAP without compromising on the solution quality. Moreover, our MFEA based solution approach has outperformed all other algorithms in terms of computation time and provides substantial improvement compared to other approaches, which establishes its better suitability to solve multiple RRAP problems together considering its at par reliability ensuring faster computation. In the ranking among all the other approaches using TOPSIS method, MFEA again shows significant performance with ranking higher for both the test sets. The statistical significance test concludes that MFEA demonstrates comparable and even improved, to a certain extent, median values, compared to other evolutionary approaches such as PSO, SA, DE, and ACO across all the case studies. Also, MFEA is statistically performing similar to other algorithms except GA, which reflects different behavior.

In MFEA, similarities between the tasks play a decisive role in finding the optimal solutions. The random mating probability (RMP) is a fixed value provided at the beginning of the execution based on some intuition of task similarities. The poor choice of RMP may lead to negative or insufficient knowledge transfer, which has a direct impact on the rate of convergence of tasks. Thus, our future work will focus on developing a new technique of automatic knowledge transfer between tasks to ensure the right amount of knowledge transfer and faster convergence in the MFEA environment without providing any prior information regarding task similarity. Furthermore, we shall apply the proposed framework in the optimization problems with bigger industrial task sets to validate the scalability. Future research may also focus on efficient modelling of uncertainty and its positive effect on the overall reliability of the system. Such uncertainty could range from the data to the system components, which when modelled with effective approaches, may result in the development of more robust and reliable systems.

Acknowledgements

Authors are sincerely thankful to the learned editors and the reviewers for all of their valuable comments, which were quite helpful in revising the manuscript, and in improving its overall technical quality. Authors are also grateful to the South Asian University, New Delhi, India for providing excellent infrastructural facilities through their Computational Intelligence Lab while carrying out this research.

References

[1] Gupta, Abhishek, Yew-Soon Ong, and Liang Feng. "Multifactorial evolution: toward evolutionary multitasking." *IEEE Transactions on Evolutionary Computation* 20, no. 3 (2015): 343-357.

[2] Wei, Tingyang, Shibin Wang, Jinghui Zhong, Dong Liu, and Jun Zhang. "A review on evolutionary multitask optimization: Trends and challenges." *IEEE Transactions on Evolutionary Computation* 26, no. 5 (2021): 941-960.

[3] Gupta, Abhishek, Lei Zhou, Yew-Soon Ong, Zefeng Chen, and Yaqing Hou. "Half a dozen real-world applications of evolutionary multitasking, and more." *IEEE Computational Intelligence Magazine* 17, no. 2 (2022): 49-66.

[4] Rausand, Marvin. *Reliability of safety-critical systems: theory and applications*. John Wiley & Sons, 2014.

[5] Garg, Harish, and S. P. Sharma. "Multi-objective reliability-redundancy allocation problem using particle swarm optimization." *Computers & Industrial Engineering* 64, no. 1 (2013): 247-255.

[6] Kuo, Way, and V. Rajendra Prasad. "An annotated overview of system-reliability optimization." *IEEE Transactions on reliability* 49, no. 2 (2000): 176-187.

[7] Coit, David W., and Enrico Zio. "The evolution of system reliability optimization." *Reliability Engineering & System Safety* 192 (2019): 106259.

[8] Huang, Chia-Ling. "A particle-based simplified swarm optimization algorithm for reliability redundancy allocation problems." *Reliability Engineering & System Safety* 142 (2015): 221-230.

[9] Jianping, Li. "A bound dynamic programming for solving reliability redundancy optimization." *Microelectronics Reliability* 36, no. 10 (1996): 1515-1520.

[10] Kuo, Way, Hsin-Hui Lin, Zhongkai Xu, and Weixing Zhang. "Reliability optimization with the Lagrange-multiplier and branch-and-bound technique." *IEEE Transactions on Reliability* 36, no. 5 (1987): 624-630.

[11] Sun, Xiaoling L., and Duan Li. "Optimality condition and branch and bound algorithm for constrained redundancy optimization in series systems." *Optimization and Engineering* 3 (2002): 53-65.

[12] Chern, Maw-Sheng. "On the computational complexity of reliability redundancy allocation in a series system." *Operations research letters* 11, no. 5 (1992): 309-315.

[13] Devi, Sarita, Harish Garg, and Deepika Garg. "A review of redundancy allocation problem for two decades: bibliometrics and future directions." *Artificial Intelligence Review* 56, no. 8 (2023): 7457-7548.

[14] Yadav, Vinay, Subhankar Karmakar, Pradip P. Kalbar, and Anil Kumar Dikshit. "PyTOPS: A Python based tool for TOPSIS." SoftwareX 9 (2019): 217-222.

[15] Chambari, Amirhossein, Parham Azimi, and Amir Abbas Najafi. "A bi-objective simulation-based optimization algorithm for redundancy allocation problem in series-parallel systems." *Expert Systems with Applications* 173 (2021): 114745.

[16] Chambari, Amirhossein, Parham Azimi, and Amir Abbas Najafi. "Redundancy allocation of heterogeneous components by simulation-based optimization." *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 235, no. 6 (2021): 1090-1101.

[17] Devi, Sarita, and Deepika Garg. "Hybrid genetic and particle swarm algorithm: redundancy allocation problem." *International Journal of System Assurance Engineering and Management* 11 (2020): 313-319.

[18] Xu, Yue, Dechang Pi, Shengxiang Yang, and Yang Chen. "A novel discrete bat algorithm for heterogeneous redundancy allocation of multi-state systems subject to probabilistic common-cause failure." *Reliability Engineering & System Safety* 208 (2021): 107338.

[19] Modibbo, Umar Muhammad, Mohd Arshad, Omer Abdalghani, and Irfan Ali. "Optimization and estimation in system reliability allocation problem." *Reliability Engineering & System Safety* 212 (2021): 107620.

[20] Yeh, Wei-Chang. "BAT-based algorithm for finding all Pareto solutions of the series-parallel redundancy allocation problem with mixed components." *Reliability Engineering & System Safety* 228 (2022): 108795.

[21] Dobani, Ehsan Ramezani, Mostafa Abouei Ardakan, Hamed Davari-Ardakani, and Mohammad N. Juybari. "RRAP-CM: A new reliability-redundancy allocation problem with heterogeneous components." *Reliability Engineering & System Safety* 191 (2019): 106563.

[22] Nath, Rahul, Zubair Ashraf, Pranab K. Muhuri, and QM Danish Lohani. "BLEAQ based solution for bilevel reliability-allocation problem." In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2661-2668. IEEE, 2017.

[23] MohammadZadeh Dogahe, Shima, and Seyed Jafar Sadjadi. "A new biobjective model to optimize integrated redundancy allocation and reliability-centered maintenance problems in a system using metaheuristics." *Mathematical Problems in Engineering* 2015 (2015).

[24] Dolatshahi-Zand, Ali, and Kaveh Khalili-Damghani. "Design of SCADA water resource management control center by a bi-objective redundancy allocation problem and particle swarm optimization." *Reliability Engineering & System Safety* 133 (2015): 11-21.

[25] Sharifi, Mani, Ali Sayyad, Sharareh Taghipour, and Abdolreza Abhari. "Optimizing a joint reliability-redundancy allocation problem with common cause multi-state failures using immune algorithm." *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 237, no. 1 (2023): 152-165.

[26] de Quadros Maia Filho, Danilo, and Michel Bessani. "Multi-objective linear approach for the reliability-redundancy allocation problem." In *2021 International Conference on Communication, Control and Information Sciences (ICCISc)*, vol. 1, pp. 1-5. IEEE, 2021.

[27] Nath, Rahul, and Pranab K. Muhuri. "Evolutionary optimization based solution approaches for many objective reliability-redundancy allocation problem." *Reliability Engineering & System Safety* 220 (2022): 108190.

[28] Nath, Rahul and., Pranab K. Muhuri. "A Novel Evolutionary Solution Approach for Many-objective Reliability-Redundancy Allocation Problem Based on Objective Prioritization and Constraint Optimization." (2023).

[29] Sharifi, M. "Solving a joint availability-redundancy optimization model with multi-state components with meta-heuristic." *International Journal of Industrial Mathematics* 12, no. 1 (2020): 59-70.

[30] Garg, Deepika, and Sarita Devi. "RAP via hybrid genetic simulating annealing algorithm." *International Journal of System Assurance Engineering and Management* 12 (2021): 419-425.

[31] Garg, Harish, Monica Rani, and S. P. Sharma. "An efficient two phase approach for solving reliability–redundancy allocation problem using artificial bee colony technique." *Computers & operations research* 40, no. 12 (2013): 2961-2969.

[32] Guilani, Pardis Pourkarim, Mostafa Abouei Ardakan, and Ehsan Ramezani Dobani. "Optimal component sequence in heterogeneous 1-out-of-N mixed RRAPs." *Reliability Engineering & System Safety* 217 (2022): 108095.

[33] Zhang, Jinchun, Hang Lv, and Jinxiu Hou. "A novel general model for RAP and RRAP optimization of k-out-of-n: G systems with mixed redundancy strategy." *Reliability Engineering & System Safety* 229 (2023): 108843.

[34] Li, Shuai, Xuefen Chi, and Baozhu Yu. "An improved particle swarm optimization algorithm for the reliability–redundancy allocation problem with global reliability." *Reliability Engineering & System Safety* 225 (2022): 108604.

[35] Sedaghat, Niloofar, and Mostafa Abouei Ardakan. "G-mixed: A new strategy for redundant components in reliability optimization problems." *Reliability Engineering & System Safety* 216 (2021): 107924.

[36] Yeh, Wei-Chang, Wenbo Zhu, Shi-Yi Tan, Gai-Ge Wang, and Yuan-Hui Yeh. "Novel general active reliability redundancy allocation problems and algorithm." *Reliability Engineering & System Safety* 218 (2022): 108167.

[37] Liu, Baoliang, Yanqing Wen, Qingan Qiu, Haiyan Shi, and Jianhui Chen. "Reliability analysis for multi-state systems under K-mixed redundancy strategy considering switching failure." *Reliability Engineering & System Safety* 228 (2022): 108814.

[38] Peiravi, Abdossaber, Mustapha Nourelfath, and Masoumeh Kazemi Zanjani. "Redundancy strategies assessment and optimization of k-out-of-n systems based on Markov chains and genetic algorithms." *Reliability Engineering & System Safety* 221 (2022): 108277.

[39] Zhang, Jianchun, Lei Li, and Zhiwei Chen. "Strength–redundancy allocation problem using artificial bee colony algorithm for multi-state systems." *Reliability Engineering & System Safety* 209 (2021): 107494.

[40] Zaretalab, Arash, Mani Sharifi, Pedram Pourkarim Guilani, Sharareh Taghipour, and Seyed Taghi Akhavan Niaki. "A multi-objective model for optimizing the redundancy allocation, component supplier selection, and reliable activities for multi-state systems." *Reliability Engineering & System Safety* 222 (2022): 108394.

[41] Zhang, Jianchun, Lei Li, and Zhiwei Chen. "Strength–redundancy allocation problem using artificial bee colony algorithm for multi-state systems." *Reliability Engineering & System Safety* 209 (2021): 107494.

[42] Shukla, Amit K., Md Abdul Malek Chowdury, Rahul Nath, and Pranab K. Muhuri. "Fuzzy reliability redundancy allocation problem using multi-factorial evolutionary algorithm." In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2512-2517. IEEE, 2021.

[43] Santiago, Alejandro, Bernabé Dorronsoro, Antonio J. Nebro, Juan J. Durillo, Oscar Castillo, and Héctor J. Fraire. "A novel multi-objective evolutionary algorithm with fuzzy logic based adaptive selection of operators: FAME." *Information Sciences* 471 (2019): 233-251.

[44] Santiago, Alejandro, Bernabe Dorronsoro, Hector J. Fraire, and Patricia Ruiz. "Micro-genetic algorithm with fuzzy selection n of operators for multi-Objective optimization: μFAME." *Swarm and Evolutionary Computation* 61 (2021): 100818.

[45] Wang, Wei, Mingqiang Lin, Yongnian Fu, Xiaoping Luo, and Hanghang Chen. "Multi-objective optimization of reliability-redundancy allocation problem for multi-type production systems considering redundancy strategies." Reliability Engineering & System Safety 193 (2020): 106681.

[46] Ardakan, Mostafa Abouei, and Mohammad Taghi Rezvan. "Multi-objective optimization of reliability–redundancy allocation problem with cold-standby strategy using NSGA-II." Reliability Engineering & System Safety 172 (2018): 225-238.

[47] Muhuri, Pranab K., and Rahul Nath. "A novel evolutionary algorithmic solution approach for bilevel reliability-redundancy allocation problem." *Reliability Engineering & System Safety* 191 (2019): 106531.

[48] Chen, Ta-Cheng. "IAs based approach for reliability redundancy allocation problems." *Applied Mathematics and Computation* 182, no. 2 (2006): 1556-1567.

[49] Garg, Harish, Monica Rani, and S. P. Sharma. "An efficient two phase approach for solving reliability–redundancy allocation problem using artificial bee colony technique." *Computers & operations research* 40, no. 12 (2013): 2961-2969.

[50] Mellal, Mohamed Arezki, and Enrico Zio. "A penalty guided stochastic fractal search approach for system reliability optimization." *Reliability Engineering & System Safety* 152 (2016): 213-227.

[51] Das, Rajib, Rahul Nath, Amit K. Shukla, and Pranab K. Muhuri. "Multi-objective Optimization Based Feature Selection Using Correlation." In *International Conference on Advanced Data Mining and Applications*, pp. 325-336. Cham: Springer Nature Switzerland, 2022.

[52] Rauniyar, Amit, Rahul Nath, and Pranab K. Muhuri. "Multi-factorial evolutionary algorithm based novel solution approach for multi-objective pollution-routing problem." *Computers & Industrial Engineering* 130 (2019): 757-771.

[53] Goldberg, David E. *Genetic algorithms*. Pearson education India, 2013.

[54] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." In *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942-1948. IEEE, 1995.

[55] Kirkpatrick, Scott, C. Daniel Gelatt Jr, and Mario P. Vecchi. "Optimization by simulated annealing." *science* 220, no. 4598 (1983): 671-680.

[56] Storn, Rainer, and Kenneth Price. "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces." *Journal of global optimization* 11 (1997): 341-359.

[57] Dorigo, Marco, Mauro Birattari, and Thomas Stutzle. "Ant colony optimization." *IEEE computational intelligence magazine* 1, no. 4 (2006): 28-39.

[58] Bali, Kavitesh Kumar, Yew-Soon Ong, Abhishek Gupta, and Puay Siew Tan. "Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II." *IEEE Transactions on Evolutionary Computation* 24, no. 1 (2019): 69-83.

[59] Gupta, Abhishek, Yew-Soon Ong, Liang Feng, and Kay Chen Tan. "Multiobjective multifactorial optimization in evolutionary multitasking." *IEEE transactions on cybernetics* 47, no. 7 (2016): 1652-1665.