Chapter 13
Optimization of System Reliability with Time-Dependent Reliability Components in Imprecise Environment Using Hybridized QPSO

Nabaranjan Bhattacharyee, Nirmal Kumar, Sanat Kumar Mahato, and Asoke Kumar Bhunia

Abstract To design reliable systems, the optimization of system reliability (SR) is a highly concerned topic in the engineering design and industry. The reliability optimization aims to develop a reliable system with higher reliability to perform satisfactorily under certain conditions and up to a specified period. This chapter considers the redundancy allocation problem as a highly non-linear and integer programming constrained optimization problem. To cope up with reality with unpredictability, we desire to consider the reliabilities of the time-dependent components that lead to a reliable time-dependent system. Further, to incorporate the fluctuating behaviour of the system's controlling parameters and uncertainty of the situations of the environment in which the system is operated, we developed the fuzzy model. As the problem is combinatorial and highly non-linear, we developed and implemented the hybridized metaheuristic technique derived by combining QPSO, a variant of particle swarm optimization, and the Big-M penalty technique to find the solution. The crisp and fuzzy (triangular and pentagonal) models are solved, and the comparative studies are presented. The statistical computations and the sensitivity studies of the HQPSO parameters are also presented corresponding to the numerical experiments.

Keywords Mission Design Life (MDL) · Redundancy Allocation Problem (RAP) · Time-varying reliability · Triangular Fuzzy Numbers (TFN) · Pentagonal Fuzzy Numbers (PtFN) · Hybridized Quantum behaved PSO (HQPSO)

N. Bhattacharyee · S. K. Mahato (✉)
Department of Mathematics, Sidho-Kanho-Birsha University, Purulia, WB 723104, India
e-mail: sanatkr_mahato.math@skbu.ac.in

N. Kumar · A. K. Bhunia
Department of Mathematics, The University of Burdwan, Bardhaman, WB 713104, India
e-mail: akbhunia@math.buruniv.ac.in

I. Ali et al. (eds.), *Computational Modelling in Industry 4.0*,
https://doi.org/10.1007/978-981-16-7723-6_13

1 Introduction

During the last few decades, many researchers have shown their keen interest in the study of reliability optimization. A wide area of applications of reliability design is observed such as engineering and industry, machine design and productions, including networking communications and transportations, etc. Also, industrialists and machine designers have been showing their interest in reliability theory and practice as it has many practical applications. Reliability analysis is an important part of many developmental works in system designing, communication systems, infrastructure development, etc. System reliability is practically the probability of successful performance of a system up to a given period under some predetermined conditions. The reliability components are taken with fixed values in most of the works reported in the literature. However, it is more realistic to consider the reliability of a system as a function of time since it undoubtedly decreases with time. In this research area, some researchers have presented such a genuine attempt in the literature review of reliability optimization. The reliability practitioners always desire to maximize the system reliability and the considered system's lifetime under certain constraints.

Several types of reliability optimization problems have been designed and solved in the literature such as Redundancy Allocation Problem (RAP), Reliability Redundancy Allocation Problem (RRAP), etc. Most attempts are found in redundancy allocation problems. In this work, our main target is to consider the RAP type of problem in which redundant components are allocated with regard to some constraints to optimize the system reliability. The renowned researchers like, Tillman et al. (1980), Sun and Li (2002), Mahapatra and Roy (2011), Mahato et al. (2020), Garg et al. (2014), Gupta et al. (2009), Mahato et al. (2013), Sahoo et al. (2013), etc. have reported important contributions in the literature. The heuristic technique in optimal reliability allocation reported by Nakagawa and Nakashima (1977), the fuzzy environment is used reliability optimization by Chen (1977), the reduced gradient method utilized by Hwang et al. (1979), a detailed study of the optimization of the reliability of a system is done by Tillman et al. (1980), the surrogate constrained algorithm is used by Nakagawa and Miyazaki (1981). The remarkable researchers such as Chern and Jan (1986), Misra (1986), Park (1987), Misra and Sharma (1991), Huang (1996), Sung and Cho (2000), Kuo et al. (2001) Sun and Li (2002), Mahapatra and Roy (2006, 2009, 2011), Gupta et al. (2009), Bhunia et al. (2010), Sahoo et al. (2010), Bhattacharyee et al. (2021), etc. are notable in the field of reliability optimization.

The uncertainty concepts such as interval, fuzzy, and intuitionistic fuzzy are introduced and studied by many researchers in the reliability analysis. These studies gave a new direction in the study of reliability models. These models are proved to be more realistic in terms of real-life phenomena. The concepts of generalized fuzzy number (Mahapatra and Roy 2011, 2012; Garg 2013; Mahato et al. 2013; Sahoo et al. 2013, 2014; Mahato et al. 2020), interval number (Bhunia and Sahoo 2011; Mahapatra and Roy 2012; Sahoo et al. 2012; Mahato et al. 2012), intuitionistic fuzzy (Garg 2013; Garg and Rani 2013; Garg et al. 2014; Garg 2015; Jamkhaneh 2017, Bhattacharyee et al. 2021) are introduced and studied in reliability theory.

The use of soft computing techniques is observed to have a great impact on reliabity optimization. Usually, the designed problems are found to be highly non-linear combinatorial problems and the analytical solutions are very difficult. So to handle such problems several soft computing algorithms are designed and implemented to solve reliability optimization problems. Soft computing techniques like, GA, PSO, and ABC, etc. are proved to be highly effective in finding the optimal reliability for any type of reliability optimization problems. The works of several researchers like, Garg (2013), Garg and Rani (2013), Khalili-Damghani et al. (2013), Garg et al. (2014a, b), Sahoo et al. (2012, 2013, 2014), Garg (2015, 2016, 2017), Gupta et al. (2009), Mahato et al. (2013, 2020), Bhattacharyee et al. (2021) are worth mentioning. The list is not exhaustive but there are lots of researchers who have developed and utilized several algorithms to solve the problems of reliability maximization.

The time-dependent reliability models are also very much relevant in reliability theory and practice. To make the models more realistic, the reliabilities of the components should be considered to be a function of time. Only a few researchers have designed the reliability models with time-dependent reliability. The works, in this regard, of Mori and Ellingwood (1993), Hamadani and Khorshidi (2013), Ganzalezet al. (2015), Hu and Mahadevan (2015), Mourelatos et al. (2015), Wang et al. (2015), Zhu and Zhifu (2016), Mostafa (2017), Ahmadivala et al. (2019), Zafar and Wang (2020), Bhattacharyee et al. (2021), etc. are noteworthy.

This chapter's main goal is to consider the reliable system having time-varying component reliabilities and the impreciseness of the environments. We have considered here the component's reliabilities to follow exponentially decreasing function of time. So, the reliability model developed here becomes time-dependent. This work has included the impreciseness in terms of triangular fuzzy and pentagonal fuzzy numbers to handle the fluctuating situations, which certainly looks to be more realistic. Hence, we have three models including the two imprecise models, viz., the crisp model, the triangular fuzzy model, and the pentagonal fuzzy model. We developed a new soft computing algorithm to solve the problems. The newly proposed algorithm is named Hybridized Quantum-behaved PSO (HQPSO) which is a variant of PSO involving the Big-M penalty technique.

Organization of the Chapter

Section 1	Introduction
Section 2	Research Gaps
Section 3	Notation and Assumptions
Section 4	Mathematical Foundations
Subsection 4.1	Relevent Definitions
Subsection 4.2	Method of Defuzzification of Fuzzy Numbers
Section 5	Problem Formulation
Subsection 5.1	The Crisp Model
Subsection 5.2	The Fuzzy Models

(continued)

(continued)

Section 6	Solution Procedure
Subsection 6.2	Particle Swarm Optimization
Subsection 6.3	Quantum behaved Particle Swarm Optimization (QPSO)
Subsection 6.4	Proposed Hybridized QPSO
Section 7	Numerical Experiments
Section 8	Result Discussions
Section 9	Acknowledgements

2 Research Gaps

It is clearly to be noted from the existing literature that most of the reliability optimization problems focused on precise environments. Some researchers have recently presented their research on imprecise environments that include interval, fuzzy, intuitionistic fuzzy, stochastic, and a mixture of these. Few works in this area are found to attempt the problems on reliability optimization using GA, PSO, hybridized PSO, ABC algorithms, Cuckoo search algorithm, and other heuristic algorithms. Moreover, in most of the works related to our paper, the reliability components are of constant values and only a few are observed to consider these as time-varying functions. A few works are also found to consider the machine design life as the objective function, and the others have taken system reliability or the cost function as the objective function.

Again, the problem's constraints are handled in several ways; only our research group in this field has incorporated the Big-M penalty method. Thus, we have been motivated eagerly to formulate a problem in reliability studies which has the machine design life as the objective function, the reliability component as exponentially decreasing functions of time, utilize the Big-M penalty technique to tackle the constraints. We use Simpson's 1/3 rule to handle the integration to get the machine design life from system reliability function and develop a hybridized Quantum-behaved Particle Swarm Optimization due to Big-M penalty method.

3 Notation and Assumptions

Throughout the chapter, we use the symbols described below. Also, the necessary assumptions to formulate the problem under consideration are given below.

3.1 Notation

Symbols	Meanings
$\tilde{P}$, $\widehat{P}$	Triangular and Pentagonal fuzzy number respectively
$\mu_{\tilde{P}}(x)$, $\mu_{\widehat{P}}(x)$	Membership function of x ∈ X w.r.t. $\tilde{P}$, $\widehat{P}$ respectively
$Cr1\left(\tilde{P}\right)$, $Cr2(\widehat{P})$	Defuzzified value of the fuzzy number $\tilde{P}$, $\widehat{P}$ respectively
$u = (u_1, u_2, \ldots, u)$	Redundancy vector (decision variable)
$R_1(u, \lambda, t)$,$\tilde{R}_2\left(u, \tilde{\lambda}, t\right)$, $\widehat{R}_3\left(u, \widehat{\lambda}, t\right)$	System reliability in crisp, triangular fuzzy and pentagonal fuzzy forms respectively
$M_1(u, \lambda, M_T)$, $\tilde{M}_2\left(u, \tilde{\lambda}, M_T\right)$, $\widehat{M}_3(u, \widehat{\lambda}, M_T)$	MDL in crisp, triangular fuzzy and pentagonal fuzzy forms respectively
$g_{1j}(u)$, $\tilde{g}_{2j}(u)$, $\widehat{g}_{3j}(u)$	Constraints usability functions in crisp, triangular fuzzy and pentagonal fuzzy environments
b_{1j},$\tilde{b}_{2j}$, $\widehat{b}_{3j}$	Availability of resources of j-th constraint in crisp, triangular fuzzy and pentagonal fuzzy environments
l_{1i}, l_{2i}	Lower bound and upper bound of u_i
S_{size}	Swarm size in QPSO
$f(p_i)$	Value of fitness function of i-th particle in its best position
$m_i^{(z)}$	Mean best position of j-thcomponent at z-th iteration
$m^{(z)}$	Mean best position vector at z-th iteration
$x_{ij}^{(z)}$:	The position of i-thparticle in the j-th swarm at z-th iteration
$A_{ij}^{(z)}$	Local attractor of the j-th component of the i-th particle at z-th iteration
Mg	Maximum number of generations
n	Dimension of the variables
F_R	Feasible region

3.2 Assumptions

To develop our proposed model, we have careful into consideration of the assumptions given here:

(i) The proposed system is a series–parallel system.
(ii) At a particular stage, the subsystem contains identical components.
(iii) Without any repair, the redundancies are always active.
(iv) Reliability of each component is an exponentially decreasing function of time.
(v) The MDL function is defined as the integral of system reliability.
(vi) Component failure in each subsystem might not be tantamount to the system to its failure.
(vii) Throughout all environments, the control parameters are well known, viz. crisp and fuzzy (triangular and pentagonal).
(viii) Fuzzy numbers (triangular and pentagonal) are of linear type.

4 Mathematical Foundations

4.1 Relevant Definitions

Definition 4.1: The fuzzy set is the pair $(x,\mu_{\tilde{P}}(x))$, where $x \in X$ and $\mu_{\tilde{P}}(x)$:X $\rightarrow$ [0, 1], X being the universe of discourse and it is represented as $\tilde{P}=\{(x,\mu_{\tilde{P}}(x))$: $x \in$ X}, where $\mu_{\tilde{P}}(x)$ denotes the membership function of $x \in X$ w.r.t. $\tilde{P}$.

Definition 4.2: The fuzzy set $\tilde{P}$ becomes convex iff $\mu_{\tilde{P}}(\lambda x_1 + (1 - \lambda)x_2) \geq \min\{\mu_{\tilde{P}}(x_1), \mu_{\tilde{P}}(x_2)\}$, for all $x_1, x_2 \in X$, where $\lambda \in [0, 1]$.

Definition 4.3: The fuzzy set $\tilde{P}$ becomes normal if $\mu_{\tilde{P}}(x) = 1$, for some $x \in X$.

Definition 4.4: A fuzzy set becomes a fuzzy number provided it is (Fig. 1).

(i) Normal
(ii) convex

The membership function of a fuzzy number $\tilde{P}$ can be described as

$$\mu_{\tilde{P}}(x) = \begin{cases} l(x), p_1 \leq x < p_2 \\ 1, \ p_2 \leq x \leq p_3 \\ u(x), p_3 < x \leq p_4 \\ 0, \ \text{otherwise.} \end{cases}$$

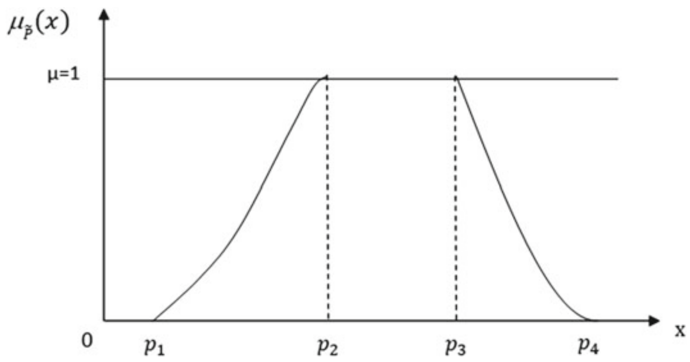

Fig. 1 General fuzzy number

$l(x)$ and $u(x)$ being the left and right shape functions, respectively.

Definition 4.5: Linear Triangular Fuzzy Number (LTFN)
An LTFN $\tilde{P}$ is represented by the triplet (p_1, p_2, p_3) and can be defined by the continuous membership function $\mu_{\tilde{P}}(x) : X \to [0, 1]$ as follows:

$$\mu_{\tilde{P}}(x) = \begin{cases} \dfrac{x - p_1}{p_2 - p_1} & \text{if } p_1 \le x \le p_2 \\ 1 & \text{if } x = p_2 \\ \dfrac{p_3 - x}{p_3 - p_2} & \text{if } p_2 \le x \le p_3 \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4.6: Linear Pentagonal Fuzzy Numbe (LPtFN)
A fuzzy pentagonal number $\widehat{P}= (p_1,p_2,p_3,p_4,p_5)$ satisfies the conditions given below:

(1) it has the continuous membership function $\mu_{\widehat{P}}(x)$ in [0,1]
(2) the membership function $\mu_{\widehat{P}}(x)$ is strictly non-decreasing in $[p_1,p_2]$ and $[p_2, p_3]$
(3) the membership function $\mu_{\widehat{P}}(x)$ is strictly non-increasing in $[p_3, p_4]$ and $[p_4, p_5]$ (Figs. 2 and 3)

4.2 Method of Defuzzification of Fuzzy Number

There are several methods of defuzzification available in the literature. The most commonly used technique for defuzzification of a fuzzy number is the centre of area (COA) method.

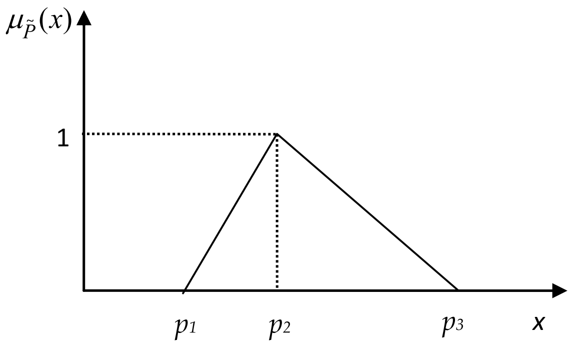

Fig. 2 Linear triangular fuzzy number

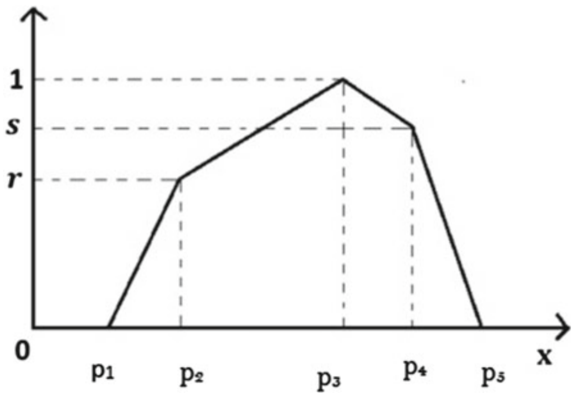

Fig. 3 Linear pentagonal fuzzy number

Let the fuzzy number $\tilde{P}$ has a continuous membership function $\mu_{\tilde{P}}(x)$ then the COA formula for crispification is defined as follows (Mahato and Bhunia 2016).

$Cr1\left(\tilde{P}\right) = \frac{\int \mu_{\tilde{P}}(x)xdx}{\int \mu_{\tilde{P}}(x)dx}$.

4.2.1 Crispification Formula for Linear Triangular Fuzzy Number

The crispification formula for Linear Triangular Fuzzy Number $\tilde{P} = (p_1, p_2, p_3)$ can be defined as (Mahato and Bhunia 2016)

$Cr1(\tilde{P}) = (p_1 + p_2 + p_3)/3$.

Example 4.1: For $\tilde{P} = (2, 3, 4)$, $p_1 = 2$, $p_2 = 3$, $p_3 = 4$, so

$$Cr1(\tilde{P}) = \tfrac{1}{3}(p_1 + p_2 + p_3)$$
$$= \tfrac{1}{3}(2 + 3 + 4)$$
$$= 3$$

Example 4.2: For $\tilde{P} = (1.6, 2.9, 3.8)$ $p_1 = 1.6$, $p_2 = 2.9$, $p_3 = 3.8$ and so

$$Cr1(\tilde{P}) = \frac{1}{3}(p_1 + p_2 + p_3)$$
$$= \frac{1}{3}(1.6 + 2.9 + 3.8)$$
$$= 2.7666666667$$

4.2.2 Crispification Formula for Linear Pentagonal Fuzzy Number

The crispification formula for Linear Pentagonal Fuzzy Number $\widehat{P} = (p_1, p_2, p_3, p_4, p_5)$ is defined as (Mahato and Bhunia 2016)

$$Cr2(\widehat{P}) = \frac{p_5^2 + p_4^2 + p_5 p_4 - p_1 p_2 - p_2^2 - p_1^2}{3(p_5 + p_4 - p_2 - p_1)}$$

Example 4.3: For $\widehat{P} = (1, 2, 3, 4, 6)$, $p_1 = 1$, $p_2 = 2$, $p_3 = 3$, $p_4 = 4$, $p_5 = 6$ and so

$$Cr2(\widehat{P}) = \frac{p_5^2 + p_4^2 + p_5 p_4 - p_1 p_2 - p_2^2 - p_1^2}{3(p_5 + p_4 - p_2 - p_1)}$$
$$= 3.2857142857$$

Example 4.4: For $\widehat{P} = (2.5, 3.3, 4.4, 5.8, 6.4)$, $p_1 = 2.5$, $p_2 = 3.3$, $p_3 = 4.4$, $p_4 = 5.8$, $p_5 = 6.4$ and so

$$Cr2(\widehat{P}) = \frac{p_5^2 + p_4^2 + p_5 p_4 - p_1 p_2 - p_2^2 - p_1^2}{3(p_5 + p_4 - p_2 - p_1)} = 4.496354$$

5 Problem Formulation

This section covers the formulation of a series–parallel reliability redundancy allocation problem using the fact that the components have time-dependent reliabilities (Bhattacharyee et al. 2021). It is supposed that the reliability components obey exponential distributions, leading to the reliability of the system being time-dependent.

Moreover, we are inspired for considering the mission design life and desire to get the maximum value of the system reliability with a proper choice of the redundancy allocation vector. Evidently, it is better not to take the controlling parameters as fixed numbers by some deterministic rule but to consider these as imprecise numbers to retain the reliable system's unpredictable nature. The parameters' estimated values cannot be predicted precisely due to the reliability system's fluctuating character. This unpredictable situation can be handled by considering the impreciseness in terms of fuzzy, intuitionistic fuzzy, interval, stochastic, or combination. In the fuzzy approach, we need to know the membership function for a given fuzzy number, while for an intuitionistic fuzzy approach, we should know both the membership function and the non-membership function. For the interval method, the parameters are taken as closed intervals. Some known probability distributions are taken in the stochastic approach.

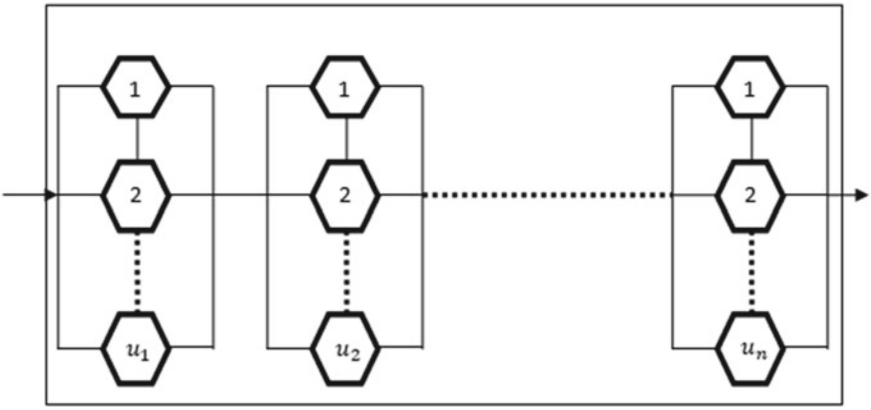

Fig. 4 Series–parallel system with n-stages

In this work, we assume the impreciseness/vagueness in the form of fuzzy numbers (TFN and PtFN). Thus, depending upon the nature of the controlling parameters, we develop three models corresponding to the series-parallelsystem (Fig. 4).

5.1 The Crisp Model

To solve the redundancy allocation problem (RAP) with time-varying reliability, we considered the reliability component following the exponential failure rate and time-dependent component reliability function. We have considered a system with n subsystems connected in series, and each subsystem consists of $u_i(i = 1, 2, \ldots, n)$ the number of active redundant components which are identical.

Let the failure density function be $f_i(t) = \lambda_i e^{-\lambda_i t}$ and the reliability function of each component of the i-th subsystem be $r_i(t) = e^{-\lambda_i t}, t > 0, \lambda_i$ is constant, $i = 1, 2, \ldots, n$.

Then using the combinatorial theory of probability, the reliability of the series–parallel system (Fig. 4) becomes $R_1(u, t; \lambda) = \prod_{i=1}^{n} \left[1 - (1 - e^{-\lambda_i t})^{u_i}\right]$.

To fulfil the aims like mission time, i.e. the system's non-stop successful functioning, cost-effectiveness, etc. the design is to be done suitably. The Mission Design Life (MDL) function is defined as (Mostafa et al. 2017; Bhattacharyee et al. 2021)

$M_1(u, \lambda, M_T) = \int_0^{M_T} R_1(u, \lambda, t)dt.$

It is easily understood that the optimization problem of maximizing the system reliability $R_1(u, \lambda, t)$ is equivalent to maximizing $M_1(u, \lambda, M_T)$. For the known parameters λ and M_T, the optimization problem can be stated as,

$$Maximize\ M_1(u, \lambda, M_T) = \int_0^{M_T} R_1(u, \lambda, t)dt$$

$$= \int_0^{M_T} \left(\prod_{i=1}^{n} \left[1 - (1 - e^{-\lambda_i t})^{u_i} \right] \right) dt$$

$$\text{subject to} \tag{1}$$

$$g_{1j}(u_1, u_2, \ldots, u_n) \le b_{1j},\ j = 1, 2, \ldots, m$$

$$l_{1i} \le y_i \le l_{2i},\ i = 1, 2, \ldots, n$$

$u = (u_1, u_2, \ldots, u_n)$ being the redundancy vector, u_i is a non-negative integer representing the redundancy level of the i-th component.

5.2 *The Fuzzy Models*

The two fuzzy models in a fuzzy environment are developed in which the control parameters are taken as linear triangular fuzzy numbers (LTFN) and linear pentagonal fuzzy numbers (LPtFN) with linear membership functions. Thus, the fuzzy models can be stated as:

5.2.1 Triangular Fuzzy Model

$$Maximize\ \underset{2}{\widetilde{M}}(u,\ \widetilde{\lambda},\ M_T) = \int_0^{M_T} R_2(u,\ \widetilde{\lambda},\ t)dt$$

$$= \int_0^{M_T} \left(\prod_{i=1}^{n} \left[1 - (1 - e^{-\widetilde{\lambda}_i t})^{u_i} \right] \right) dt$$

$$\text{subject to} \tag{2}$$

$$\tilde{g}_{2j}(u_1, u_2, \ldots, u_n) \le \tilde{b}_{2j},\ j = 1, 2, \ldots, m$$

$$l_{1i} \le u_i \le l_{2i},\ i = 1, 2, \ldots, n$$

$u = (u_1, u_2, \ldots, u_n)$ being the redundancy vector, u_i is a nonnegative integer representing the redundancy level of the i-th component.

5.2.2 Pentagonal Fuzzy Model

$$
\begin{aligned}
Maximize\ \hat{M}_2(u,\ \hat{\lambda},\ M_T) &= \int_0^{M_T} \hat{R}_3(u,\ \hat{\lambda},\ t)dt \\
&= \int_0^{M_T} \left(\prod_{i=1}^{n} \left[1 - (1 - e^{-\hat{\lambda}_i t})^{u_i} \right] \right) dt
\end{aligned}
$$

subject to (3)

$$\widehat{g}_{3j}(u_1, u_2, \ldots, u_n) \le \widehat{b}_{3j},\ j = 1, 2, \ldots, m$$

$$l_{1i} \le u_i \le l_{2i},\ i = 1, 2, \ldots, n$$

$u = (u_1, u_2, \ldots, u_n)$ being the redundancy vector, u_i is a non-negative integer representing the redundancy level of the *i*-th component.

6 Solution Procedure

The objective functions in problems (1), (2) and (3) all are highly non-linear and the problems are combinatorial optimization problems. The objective functions are to be maximized that involve the integration of the system reliability of a series–parallel system. The integration is quite difficult to evaluate analytically so the Simpson's 1/3 rule is utilized to evaluate the approximate integral value.

6.1 Particle Swarm Optimization (PSO)

Kennedy and Eberhart (1995) reported the new algorithm as being inspired by the social behaviours of fish schooling and birds flocking. This is known as the PSO algorithms and proved to be efficient enough in solving global optimization problems. In this algorithm, every solution of the swarm is represented as bird/fish like particles and they have the liberty to fly throughout the solution space with the common goal to land on or near of the optimal position. The position of each particle is updated by the combined knowledge of the individual and the group of the swarm. Each particle remembers its personal best (*pbest*) along with the group best position or global best (*gbest*).

Let us take,

$x_i^{(z)} = \left(x_{i1}^{(z)}, x_{i2}^{(z)}, \ldots, x_{in}^{(z)}\right)$, as the current position

$v_i^z = \left(v_{i1}^{(z)}, v_{i2}^{(z)}, \ldots, v_{in}^{(z)}\right)$, as the current velocity

$p_i^z = \left(p_{i1}^{(z)}, p_{i2}^{(z)}, \ldots, p_{in}^{(z)}\right)$, as the *pbest* position

$p_g^{(z)} = \left(p_{g1}^{(z)}, p_{g2}^{(z)}, \ldots, p_{gn}^{(z)}\right)$, as the *gbest* position respectively at the z-th iteration of the i-th swarm.

Then the updation formulae for the velocity and position of the i-th particle in the j-th direction at the z-th iteration are given by

$$v_{ij}^{(z+1)} = v_{ij}^{(z)} + c_1 r_{1j}^{(z)}\left(p_{ij}^{(z)} - x_{ij}^{(z)}\right) + c_2 r_{2j}^{(z)}\left(p_{gj}^{(z)} - x_{ij}^{(z)}\right) \tag{5}$$

$$x_{ij}^{(z+1)} = x_{ij}^{(z)} + v_{ij}^{(z+1)} \tag{6}$$

where $i = 1, 2, \ldots, S_{size}; j = 1, 2, \ldots, n; z = 1, 2, \ldots, Mg; c_1(> 0), c_2(> 0)$ are the acceleration coefficients and $r_{1j}^{(z)}, r_{2j}^{(z)} \sim U(0, 1)$.

6.2 Quantum Behaved Particle Swarm Optimization (QPSO)

The strategies done by traditional PSO completely fail in quantum space because the velocity and position cannot be specified concurrently according to '*Heisenberg's Uncertainty Principle*'. So it needs to describe the particles in terms of the wave function. While moving in the quantum space, the wave function $\psi(x, t)$ must satisfy the Schrödinger wave equation and by solving the equation, we get the density function $|\psi|^2$. Utilizing the Monte Carlo technique, the updating formula is obtained as stated below

$$x_{ij}^{(z+1)} = A_{ij}^{(z)} + \beta\left|x_{ij}^{(z)} - m_j^{(z)}\right| \log\left(\frac{1}{u_{ij}^{(z)}}\right) \quad if \;\; r \geq 0.5 \tag{7}$$

$$x_{ij}^{(z+1)} = A_{ij}^{(z)} - \beta\left|x_{ij}^{(z)} - m_j^{(z)}\right| \log\left(\frac{1}{u_{ij}^{(z)}}\right) \quad if \;\; r < 0.5 \tag{8}$$

where $A_{ij}^{(z)} = \phi_j p_{ij}^{(z)} + (1 - \phi_j) p_{gj}^{(z)}$,

$m^{(z)} =$ averages of all *pbest* positions

$$= \left(m_1^{(z)}, m_2^{(z)}, \ldots, m_n^{(z)}\right)$$

$$= \left(\frac{1}{S_{size}} \sum_{i=1}^{S_{size}} p_{i1}^{(z)}, \frac{1}{S_{size}} \sum_{i=1}^{S_{size}} p_{i2}^{(z)}, \ldots, \frac{1}{S_{size}} \sum_{i=1}^{S_{size}} p_{in}^{(z)}\right) \tag{9}$$

$\beta =$ expansion contraction parameter

$u_{ij}^{(z)}$, r are random numbers in (0,1).

6.3 Proposed Hybridized QPSO (HQPSO)

In order to use the QPSO after combing with the Big-M penalty, we have modified the QPSO algorithm and developed the hybrid form of it. This hybrid algorithm combines the features of the QPSO and the Big-M penalty technique. The Big-M penalty function techniques has the capability to wipe out the infeasible solutions from the search region reducing the constrained optimization into unconstrained one. This is similar to the notion that the particles will never search for food in the points once observed not to contain any food. We have developed the HQPSO especially to solve the pure integer programming problems of combinatorial type involving the integration of highly non-linear integrand. The constrained optimization problems can easily be solved by implementing this new HQPSO. In the Big-M penalty method, a very big/small value is assigned as the fitness value corresponding to the infeasible points/positions according to the problem (minimization/maximization). In this method, the infeasible points/positions are never revisited and so the efficiency of the algorithm increases with quick convergence in the feasible region

$$F_R = \left\{u = (u_1, u_2, ..., u_n) : g_{1j}(u_1, u_2, ..., u_n) \le b_{1j};\ j = 1, 2, ..., n\right\}.$$

The iterative steps of HQPSO are given.

Step 1: Start
Step 2: Initialize QPSO parameters and also the bounds of the variables
Step 3: Create a random particles' swarm, i.e. randomly generate
$X_{ij}(\ i = 1(1)S_{size};\ j = 1(1)n)$
Step 4: Set this initial positions as ***pbest*** *position i.e.* $p_i = x_i$ *for* $i = 1,2,\ldots, S_{size}$
Step 5: Determine ***gbest*** *position,* $g = arg(max\ (f(p_i)))$ *for* $i = 1(1)\ S_{size}$
Step 6: Set $z = 1$
Step 7: Calculate mean best position m using Eq. (9)
Step 8: Generate $\phi = rand(0, 1)$
Step 9: Compute local attractor $A_{ij} = \phi p_{ij} + (1 - \phi)p_{gj}$
Step 10: Generate $r = rand(0, 1)$
Step 11: If $r > 0.5$, $x_{ij} = A_{ij} + \beta\left|m_j - x_{ij}\right| ln(\frac{1}{u_{ij}})$
Step 12: Otherwise, $x_{ij} = A_{ij} - \beta\left|m_j - x_{ij}\right| ln(\frac{1}{u_{ij}})$
Step 13: If $f(x_i) \in F_R$ *assign* $f(x_i) = -M$
Step 14: If $f\ (p_i) < f(x_i)$, *set* $p_i = x_i$
Step 15: Otherwise, $g = arg(max(f(p_i)))$
Step 16: if $z < Mg$, $z = z + 1$ *and follow Step 7*
Step 17: Otherwise, print the result
Step 18: Stop.

7 Numerical Experiments

For the illustration of the methodology, we have considered three numerical examples given below (Bhattacharyee et al. 2021). These examples are provided with crisp data. The input data for the triangular and pentagonal fuzzy models can be found in Tables 1, 2, 3, 4, 5 and 6, respectively. The defuzzified data are computed by the formulae described in Sect. 4.2 (Tables 7, 8, 9, 10, 11 and 12).

Example 1: Crisp Form

$$MaximizeM_1(u, \lambda, M_T) = \int_0^{M_T} R_1(u, \lambda, t)dt$$

$$where\ \ R_1(u,\ \lambda\, t)\ =\ \left(\prod_{i=1}^{4}[1-(1-e^{-\lambda_i t})^{u_i}]\right)$$

subject to

$$C_s\ =\ \sum_{i=1}^{4} c_i u_i\ \leq\ C$$

$$W_s\ =\ \sum_{i=1}^{4} w_i u_i\ \leq\ W$$

Table 1 Data for Example 1 (LTFN)

i	$\tilde{\lambda}_i\ (\times 10^{-4})$	$\tilde{c}_i$	$\tilde{w}_i$
1	(9.415,9.431,9.446)	(0.2,1.2,2.4)	(3,5,6)
2	(5.118,5.129,5.138)	(2.0,2.3,2.8)	(2,4,5)
3	(8.323,8.338,8.349)	(3.0,3.4,3.9)	(6,8,9)
4	(16.151,16.252,16.354)	(4.0,4.5,4.8)	(5,7,8)
$\tilde{C}$= (50,56,60); $\tilde{W}$= (113,120,125)			

Table 2 Data for Example 1 (LPtFN)

i	$\widehat{\lambda}_i(\times 10^{-4})$	$\widehat{c}_i$	$\widehat{w}_i$
1	(9.102,9.415,9.431,9.446,9.521)	(0.1,0.2,1.2,2.4,2.6)	(2,3,5,6,8)
2	(5.108,5.118,5.129,5.138,5.141)	(1.8,2.0,2.3,2.8,2.9)	(1,2,4,5,6)
3	(8.320,8.323,8.338,8.349,8.356)	(2.7,3.0,3.4,3.9,4.1)	(4,6,8,9,10)
4	(16.143,16.151,16.252,16.354,16.361)	(3.7,4.0,4.5,4.8,4.9)	(3,5,7,8,10)
$\widehat{C}$= (48,50,56,60,63); $\widehat{W}$= (111,113,120,125,129)			

Table 3 Data for Example 2 (LTFN)

i	$\tilde{\lambda}_i$ $(\times 10^{-4})$	$\tilde{v}_i$	$\tilde{c}_i$	$\tilde{w}_i$
1	(9.415,9.431,9.446)	(0.15,1.0,1.48)	(6.4,7.0,7.7)	(6.7,7.0,7.6)
2	(5.118,5.129,5.138)	(1.4,2.0,2.7)	(6.5,7.0,7.8)	(7.8,8.0,8.4)
3	(8.323,8.338,8.349)	(2.4,3.0,3.8)	(4.8,5.0,5.6)	(7.5,8.0,8.3)
4	(16.151,16.252,16.354)	(3.6,4.0,4.8)	(8.3,9.0,9.5)	(5.1,6.0,6.7)
5	(7.147,7.257,7.361)	(1.6,2.0,2.7)	(3.4,4.0,4.7)	(8.7,9.0,9.7)
$\tilde{V}$= (103,110,123); $\tilde{C}$ = (167, 175, 188); $\tilde{W}$= (178,200,227)				

$$u_i \in Z^+, r_i(t) = e^{-\lambda_i t}, i = 1, 2, 3, 4; M_T = 100hrs,$$

$$C = 56, c = (c_1, c_2, c_3, c_4) = (1.2, 2.3, 3.4, 4.5),$$

$$W = 120, w = (w_1, w_2, w_3, w_4) = (5, 4, 8, 7)$$

$$\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (0.0009431, 0.0005129, 0.0008338, 0.0016252).$$

Example 2: Crisp Form

$$Maximize M_1(u, \lambda, M_T) = \int_0^{M_T} R_1(u, \lambda, t)dt$$

$$where\ \ R_1(u,\ \lambda\, t)\ =\ \left(\prod_{i=1}^{5}\left[1 - (1 - e^{-\lambda_i t})^{u_i}\right]\right)$$

subject to

$$V_s\ =\ \sum_{i=1}^{5} v_i u_i^2\ \leq\ V$$

$$C_s\ =\ \sum_{i=1}^{5} c_i[u_i\ +\ e^{\frac{u_i}{4}}]\ \leq\ C$$

$$W_s\ =\ \sum_{i=1}^{5} w_i[u_i e^{\frac{u_i}{4}}]\ \leq\ W$$

$$u_i \in Z^+, r_i(t) = e^{-\lambda_i t}, i = 1, 2, 3, 4, 5; M_T = 100hrs,$$

Table 4 Data for Example 2 (LPtFN)

i	$\widehat{\lambda}_i(\times 10^{-4})$	$\widehat{v}_i$	$\widehat{c}_i$	$\widehat{w}_i$
1	(9.408,9.415,9.431,9.446,9.449)	(0.13,0.15,1.0,1.48,1.50)	(6.3,6.4,7.0,7.7,7.9)	(6.5,6.7,7.0,7.6,7.9)
2	(5.109,5.118,5.129,5.138,5.140)	(1.2,1.4,2.0,2.7,2.9)	(6.3,6.5,7.0,7.8,7.9)	(7.7,7.8,8.0,8.4,8.6)
3	(8.321,8.323,8.338,8.349,8.350)	(2.2,2.4,3.0,3.8,3.9)	(4.5,4.8,5.0,5.6,5.8)	(7.3,7.5,8.0,8.3,8.6)
4	(16.149,16.151,16.252,16.354,16.357)	(3.5,3.6,4.0,4.8,4.9)	(8.1,8.3,9.0,9.5,9.8)	(5.0,5.1,6.0,6.7,6.9)
5	(7.141,7.147,7.257,7.361,7.364)	(1.3,1.6,2.0,2.7,2.9)	(3.1,3.4,4.0,4.7,4.9)	(8.4,8.7,9.0,9.7,9.9)

$\widehat{V}=$ (101,103,110,123,125); $\widehat{C}=$ (160, 167, 175, 188, 194); $\widehat{W}$=(171,178,200,227,234)

Table 5 Data for Example 3 (LTFN)

i	$\widetilde{\lambda}_i$ $(\times 10^{-4})$	$\widetilde{c}_i$	$\widetilde{w}_i$
1	(9.415,9.431,9.446)	(3,5,6)	(2,5,6)
2	(5.118,5.129,5.138)	(2,4,7)	(3,4,7)
3	(8.323,8.338,8.349)	(8,9,11)	(7,9,10)
4	(16.151,16.252,16.354)	(5,7,8)	(6,7,8)
5	(7.147,7.257,7.361)	(4,7,9)	(5,7,10)
6	(2.015,2.020,2.027)	(4,5,8)	(4,5,7)
7	(9.422,9.431,9.448)	(5,6,9)	(3,6,8)
8	(21.054,21.072,21.083)	(7,9,10)	(6,9,11)
9	(4.071,4.082,4.093)	(2,4,5)	(2,4,5)
10	(16.241,16.252,16.354)	(4,5,7)	(4,5,7)
11	(6.173,6.188,6.197)	(5,6,8)	(5,6,8)
12	(23.562,23.572,23.579)	(4,7,8)	(5,7,8)
13	(1.001,1.005,1.016)	(7,9,10)	(6,9,10)
14	(5.109,5.129,5.137)	(6,8,11)	(7,8,11)
15	(23.561,23.572,23.583)	(3,6,8)	(4,5,7)
$\tilde{C}$= (376,400,460); $\tilde{W}$= (375,414,475)			

Table 6 Data for Example 3 (LPtFN)

i	$\widehat{\lambda}_i (\times 10^{-4})$	$\widehat{c}_i$	$\widehat{w}_i$
1	(9.414,9.415,9.431,9.446,9.449)	(1,3,5,6,7)	(1,2,5,6,7)
2	(5.108,5.118,5.129,5.138,5.140)	(1,2,4,7,8)	(2,3,4,7,8)
3	(8.320,8.323,8.338,8.349,8.354)	(7,8,9,11,12)	(5,7,9,10,12)
4	(16.142,16.151,16.252,16.354,16.359)	(3,5,7,8,9)	(4,6,7,9,10)
5	(7.144,7.147,7.257,7.361,7.363)	(3,4,7,9,11)	(4,5,7,10,12)
6	(2.008,2.015,2.020,2.027,2.029)	(2,4,5,8,11)	(3,4,5,7,9)
7	(9.415,9.422,9.431,9.448,9.451)	(1,5,6,9,10)	(1,3,6,8,10)
8	(21.049,21.054,21.072,21.083,21.085)	(5,7,9,10,12)	(4,6,9,11,12)
9	(4.068,4.071,4.082,4.093,4.095)	(1,2,4,5,7)	(1,2,4,5,7)
10	(16.235,16.241,16.252,16.354,16.359)	(1,4,5,7,9)	(3,4,5,7,8)
11	(6.171,6.173,6.188,6.197,6.199)	(4,5,6,8,9)	(3,5,6,8,9)
12	(23.555,23.562,23.572,23.579,23.582)	(3,4,7,8,10)	(4,5,7,8,9)
13	(1.000,1.001,1.005,1.016,1.018)	(5,7,9,10,11)	(4,6,9,10,11)
14	(5.107,5.109,5.129,5.137,5.139)	(4,6,8,11,12)	(4,7,8,11,13)
15	(23.559,23.561,23.572,23.583, 23.585)	(1,3,6,8,11)	(3,4,5,7,9)
$\widehat{C}$= (371,376,400,460,465); $\widehat{W}$= (368,375,414,475,482)			

Table 7 Crispified data for Example 1 (LTFN)

i	$\tilde{\lambda}_i$ $(\times 10^{-4})$	$\tilde{c}_i$	$\tilde{w}_i$
1	9.430667	1.266667	4.666667
2	5.128333	2.366667	3.666667
3	8.336667	3.433333	7.666667
4	16.25233	4.433333	6.666667
$\tilde{C}$= 55.33333; $\tilde{W}$= 119.3333			

Table 8 Crispified data for Example 1 (LPtFN)

i	$\hat{\lambda}_i (\times 10^{-4})$	$\hat{c}_i$	$\widehat{w}_i$
1	9.353899	1.325532	4.777778
2	5.126107	2.373684	3.50000
3	8.337054	3.423188	7.222222
4	16.25225	4.346667	6.50000
$\widehat{C}$= 55.26667; $\widehat{W}$= 119.533			

Table 9 Crispified data for Example 2 (LTFN)

i	$\tilde{\lambda}_i$ $(\times 10^{-4})$	$\tilde{v}_i$	$\tilde{c}_i$	$\tilde{w}_i$
1	9.430667	0.876667	7.033333	7.10000
2	5.128333	2.033333	7.10000	8.066667
3	8.336667	3.066667	5.133333	7.933333
4	16.25233	4.133333	8.933333	5.933333
5	7.25500	2.10000	4.033333	9.133333
$\tilde{V}$= 112; $\tilde{C}$ = 176.6667; $\tilde{W}$= 201.6667				

Table 10 Crispified data for Example 2 (LPtFN)

i	$\hat{\lambda}_i (\times 10^{-4})$	$\hat{v}_i$	$\hat{c}_i$	$\widehat{w}_i$
1	9.429454	0.81500	7.075862	7.176812
2	5.126124	2.05000	7.124138	8.126667
3	8.335745	3.074194	5.173016	7.926984
4	16.25275	4.2	8.926437	5.925714
5	7.253245	2.123457	4.023656	9.173333
$\widehat{V}$=113; $\widehat{C}$ = 177.2303; $\widehat{W}$= 202.5				

Table 11 Crispified data for Example 3 (LTFN)

i	$\tilde{\lambda}_i$ $(\times 10^{-4})$	$\tilde{c}_i$	$\tilde{w}_i$
1	9.430667	4.666667	4.333333
2	5.128333	4.333333	4.666667
3	8.336667	9.333333	8.666667
4	16.25233	6.666667	7.333333
5	7.25500	6.666667	7.333333
6	2.020667	5.666667	5.333333
7	9.433667	6.666667	5.666667
8	21.06967	8.666667	8.666667
9	4.08200	3.666667	3.666667
10	16.28233	5.333333	5.333333
11	6.18600	6.333333	6.333333
12	23.5710	6.333333	6.666667
13	1.007333	8.666667	8.333333
14	5.12500	8.333333	8.666667
15	23.57200	5.666667	5.333333
$\tilde{C}= 412$; $\tilde{W}= 421.3333$			

Table 12 Crispified data for Example 3 (LPtFN)

i	$\hat{\lambda}_i (\times 10^{-4})$	$\hat{c}_i$	$\widehat{w}_i$
1	9.43101	2.222222	4.000000
2	5.125846	4.500000	5.000000
3	8.336522	9.500000	8.500000
4	16.25149	6.222222	7.222222
5	7.253749	6.769231	7.769231
6	2.019636	6.282051	5.777778
7	9.433946	6.153846	5.500000
8	21.06772	8.500000	8.230769
9	4.081741	3.777778	3.777778
10	16.29725	4.666667	5.500000
11	6.185000	6.500000	6.222222
12	23.56942	6.272727	6.500000
13	1.008758	8.222222	7.727273
14	5.123000	8.230769	8.717949
15	23.57200	5.777778	5.777778
$\widehat{C} = 418$; $\widehat{W} = 425$			

$$V = 110, v = (v_1, v_2, v_3, v_4, v_5) = (1, 2, 3, 4, 2),$$

$$C = 175, c = (c_1, c_2, c_3, c_4, c_5) = (7, 7, 5, 9, 4),$$

$$W = 200,\ w = (w_1, w_2, w_3, w_4, w_5) = (7, 8, 8, 6, 9)$$

$$\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) = (0.0009431, 0.0005129, 0.0008338, 0.0016252, 0.0007257).$$

Example 3: Crisp Form

$$Maximize M_1(u, \lambda, M_T) = \int_0^{M_T} R_1(u, \lambda, t)dt$$

$$where\ \ R_1(u,\ \lambda\, t)\ =\ \left(\prod_{i=1}^{15}\left[1 - (1 - e^{-\lambda_i t})^{u_i}\right]\right)$$

subject to

$$C_s\ =\ \sum_{i=1}^{15} c_i u_i\ \le\ C$$

$$W_s = \sum_{i=1}^{15} w_i u_i \le W$$

$$u_i \in Z^+, r_i(t) = e^{-\lambda_i t}, i = 1, 2, \ldots, 15, M_T = 100hrs, C = 400,$$
$$c = (c_1, c_2, \ldots, c_{15}) = (5, 4, 9, 7, 7, 5, 6, 9, 4, 5, 6, 7, 9, 8, 6),$$

$$W = 414, w = (w_1, w_2, \ldots, w_{15}) = (5, 4, 9, 7, 7, 5, 6, 9, 4, 5, 6, 7, 9, 8, 5)$$

$$\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{15}) = \begin{pmatrix} 0.0009431, 0.0005129, 0.0008338, 0.0016252, 0.0007257, \\ 0.0002020, 0.0009431, 0.0021072, 0.0004082, 0.0016252, \\ 0.0006188, 0.0023572, 0.0001005, 0.0005129, 0.0023572 \end{pmatrix}.$$

8 Result Discussions

Here, we have formulated and solved three numerical experiments for testing the efficiency of our proposed method to maximize the MDL as well as to maximize reliability of the system under optimal redundancies. In this work, we have executed 30 independent runs for each numerical problem with the help of HQPSO algorithm coded in C + + in a notebook with Intel i3 processor, 4 GB RAM in Linux operating system. To study the robustness, we have procured the results to identify the best and worst values of MDL, its average value, standard deviation and the execution time along with the corresponding system reliability. In this HWQPSO, the population size and the maximum number of generations for the three experiments are taken respectively as 70, 100; 80, 150 and 300, 700.

From Tables 13, 14, and 15, we can see the results of problems 1, 2, and 3, respectively. It is evident that the respective standard deviations are 0, 0, and 0.0023557. Figures 5, 6 and 7, respectively, show convergence history of the objective functions (MDL) in crisp form as stable w.r.t. the number of generations in wide range. Table 16 presents the comparative results of Example 1 in crisp, triangular fuzzy, and pentagonal fuzzy cases indicating that it achieved the best result in PtFN case. From Table 17, the comparative results of Example 2 can be seen and it is noticed that the best result corresponds to PtFN case. The comparative results of Example 3 are given in Table 18 indicating that the best output is obtained in PtFN case.

Table 13 Statistical results for a crisp form of Example 1 (S_{size}-70, Maxgen-100)

Best	Worst	Average	Standard deviation	Average running time
99.9987559819	99.9987559819	99.9987559819	0	0.068688

Table 14 Statistical results for a crisp form of Example 2 (S_{size}-80, Maxgen-150)

Best	Worst	Average	Standard deviation	Average running time
98.4880171489	98.4880171489	98.4880171489	0	0.0729564

Table 15 Statistical results for a crisp form of Example 3 (S_{size}-100, Maxgen-200)

Best	Worst	Average	Standard deviation	Average running time
99.9774303072	99.9684707239	99.97656270762	0.0023557	5.623954

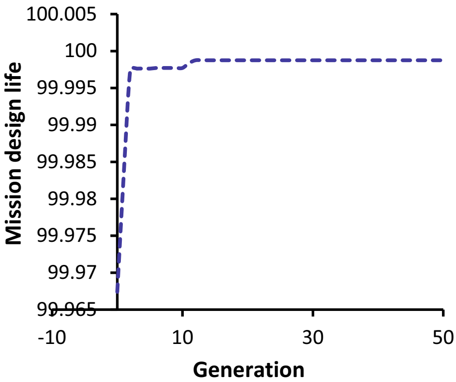

Fig. 5 Convergence history of MDL of Example 1 (crisp form) using HQPSO

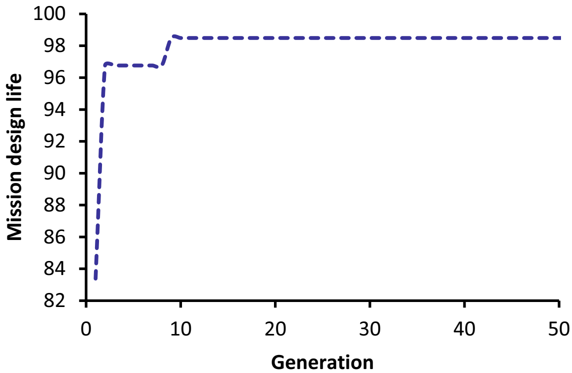

Fig. 6 Convergence history of MDL of Example 2 (crisp form) using HQPSO

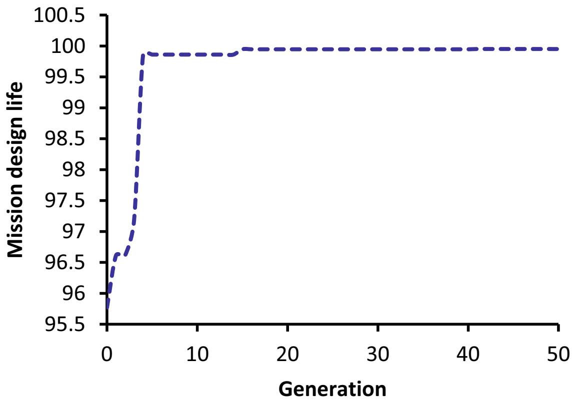

Fig. 7 Convergence history of MDL of Example 3(crisp form) using HQPSO

Table 16 Comparative results for Example 1

i	R_i	u_i	MDL^*	SR^*	CPU Time (s)
Crisp Case					
1	0.999994	5	99.9987559819	0.9999355000	0.069200
2	0.999994	4			
3	0.999959	4			
4	0.999989	6			
Triangular Fuzzy Case					
1	0.999994	5	99.9987565642	0.9999355280	0.054976
2	0.999994	4			
3	0.999959	4			
4	0.999989	6			
Pentagonal Fuzzy Case					
1	0.999994	5	**99.9987605507**	**0.999935757**	**0.051477**
2	0.999994	4			
3	0.999959	4			
4	0.999989	6			

Table 17 Comparative results for Example 2

i	R_i	u_i	MDL^*	SR^*	CPU Time (s)
Crisp Case					
1	0.991900	2	98.4880171489	0.9562601906	0.068281
2	0.997500	2			
3	0.993600	2			
4	0.977500	2			
5	0.995100	2			
Triangular Fuzzy Case					
1	0.991901	2	98.4881801367	0.9562648909	0.071150
2	0.997501	2			
3	0.993602	2			
4	0.977499	2			
5	0.995103	2			
Pentagonal Fuzzy Case					
1	0.991903	2	**98.4884012663**	**0.9562712705**	**0.072423**
2	0.997503	2			
3	0.993604	2			
4	0.977498	2			
5	0.995105	2			

Table 18 Comparative results for Example 3

i	R_i	u_i	MDL^*	SR^*	CPU Time (s)
Crisp Case					
1	0.999934	4	99.9774303072	0.9989171631	6.184960
2	0.999994	4			
3	0.999959	4			
4	0.999924	5			
5	0.999976	4			
6	0.999992	3			
7	0.999934	4			
8	0.999752	5			
9	0.999936	3			
10	0.999924	5			
11	0.999987	4			
12	0.999914	6			
13	0.999900	2			
14	0.999875	3			
15	0.999914	6			
Triangular Fuzzy Case					
1	0.999934	4	99.9811300948	0.9991167640	4.947174
2	0.999994	4			
3	0.999959	4			
4	0.999924	5			
5	0.999976	4			
6	0.999992	3			
7	0.999934	4			
8	0.999753	6			
9	0.999936	3			
10	0.999923	5			
11	0.999987	4			
12	0.999914	6			
13	0.999900	2			
14	0.999875	3			
15	0.999914	6			
Pentagonal Fuzzy Case					
1	0.999934	4	**99.9890618878**	**0.9993957023**	**4.844208**
2	0.999994	4			
3	0.999959	4			

(continued)

Table 18 (continued)

i	R_i	u_i	MDL^*	SR^*	CPU Time (s)
4	0.999924	5			
5	0.999976	4			
6	0.999992	3			
7	0.999934	4			
8	0.999953	6			
9	0.999997	4			
10	0.999923	5			
11	0.999987	4			
12	0.999914	6			
13	0.999999	3			
14	0.999994	4			
15	0.999914	6			

9 Conclusions and Future Directions

This work explores a more realistic and practical form of redundancy allocation problem where time-dependent reliabilities for the components in decreasing exponential function are considered. We use the mission design life (MDL) as the objective function rather than the traditional system reliability. Integrating the system reliability between zero (0) and mission time (M_T), the MDL is obtained. The objective function, MDL is then maximized along with the system reliability under optimal redundancy allocations (Fig. 8 and Tables 19, 20, 21).

Table 19 Best found solution of Example 1 (LPtFN case). [Popsize-70; Maximum generations-100]

x_1	x_2	x_3	x_4	R_1	R_2	R_3	R_4	Mission Design Life	System Reliability
5	4	4	6	0.999994	0.999994	0.999959	0.999989	99.9987605507	0.999935757

Table 20 Worst found a solution of Example 1 (LPtFN case). [Popsize-70; Maximum generations-100]

x_1	x_2	x_3	x_4	R_1	R_2	R_3	R_4	Mission Design Life	System Reliability
5	4	4	6	0.999994	0.999994	0.999959	0.999989	99.9987605507	0.999935757

Table 21 Statistical data for Example 1 (LPtFN case)

Best	Worst	Average	Standard deviation	Average running time
99.9987605507	99.9987605507	99.9987605507	0	0.064034

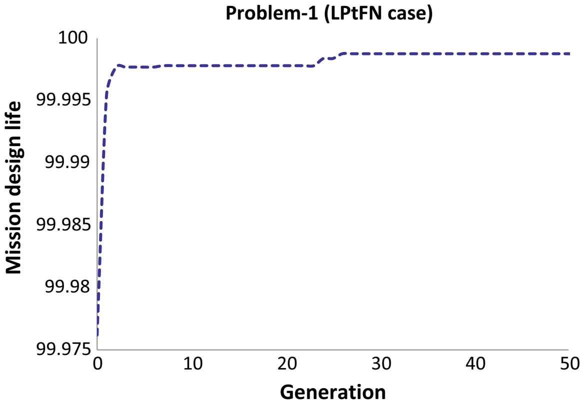

Fig. 8 Convergence history of MDL of Example 1 (LPtFN case) using HQPSO

The two fuzzy models (triangular fuzzy and pentagonal fuzzy) are developed to show the effects of uncertainty along with the crisp one of the reliability system. To evaluate the MDL as an integral of the quite complex integral, Simpson's 1/3 rule is utilized. A new PSO algorithm is developed by combining the characteristics of QPSO and the Big-M penalty technique. This hybridized algorithm HQPSO is implemented for solving the three numerical examples under consideration in three different forms, namely crisp, triangular fuzzy, and pentagonal fuzzy. The performance of the proposed algorithm is well established in these experiments (Fig. 9 and Tables 22, 23, 24).

Table 22 Best found solution of Example 2 (LPtFN case). [Popsize-80; Maximum generations-150]

x_1	x_2	x_3	x_4	x_5	R_1	R_2	R_3	R_4	R_5	Mission Design Life	System Reliability
2	2	2	2	2	0.991903	0.997503	0.993604	0.977498	0.995105	98.4884012663	0.9562712705

Table 23 Worst found a solution of Example 2 (LPtFN case). [Popsize-80; Maximum generations-150]

x_1	x_2	x_3	x_4	x_5	R_1	R_2	R_3	R_4	R_5	Mission Design Life	System Reliability
2	2	2	2	2	0.991903	0.997503	0.993604	0.977498	0.995105	98.4884012663	0.9562712705

Table 24 Statistical data for Example 2 (LPtFN case)

Best	Worst	Average	Standard deviation	Average running time
98.4884012663	98.4884012663	98.4884012663	0	0.084681

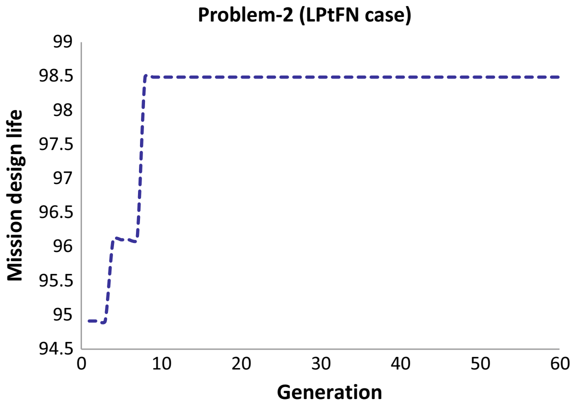

Fig. 9 Convergence history of MDL of Example 2 (LPtFN case) using HQPSO

The proposed methodology for maximizing MDL can be applied in the field of reliability optimization, system design, engineering design, industrial problems, etc. Soft computing techniques like GA, ABC algorithm, DE, Taboo search, Cuckoo search, Neural Network, Tournament-based PSO, etc. can be employed to solve this kind of problem. To consider the uncertainty, several other imprecise environments can be considered (Fig. 10 and Tables 25, 26, 27).

Table 25 Best found solution of Example 3 (LPtFN case). [Popsize-300; Maximum generations-700]

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
4	4	4	5	4	3	4	6	4	5	4	6	3	4	6

R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
0.999934	0.999994	0.999959	0.999924	0.999976	0.999992	0.999934	0.999953	0.999997	0.999923

R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	Mission Design Life	System Reliability
0.999987	0.999914	0.999999	0.999994	0.999914	99.9890618878	0.9993957023

Table 26 Worst found solution of Example 3 (LPtFN case). [Popsize-300; Maximum generations-700]

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
5	4	4	5	4	3	5	5	4	6	4	6	3	3	6

R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
0.999994	0.999994	0.999959	0.999924	0.999976	0.999992	0.999994	0.999753	0.999997	0.999988

R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	Mission Design Life	System Reliability
0.999987	0.999914	0.999999	0.999875	0.999914	99.9859984351	0.9992618745

Table 27 Statistical data for Example 3 (LPtFN case)

Best	Worst	Average	Standard deviation	Average running time
99.9890618878	99.9859984351	99.9885610794	0.000616189	5.91237

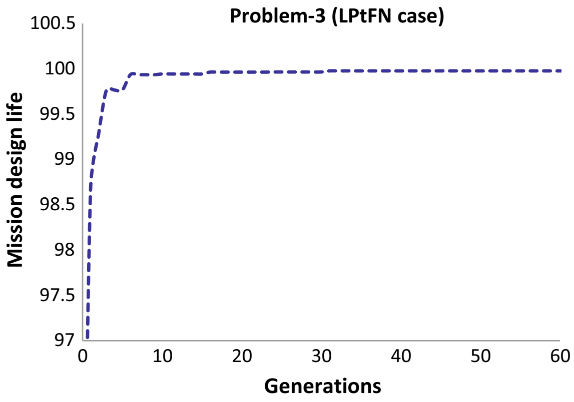

Fig. 10 Convergence history of MDL of Example 3 (LPtFN case) using HQPSO

Acknowledgements The authors are thankful to the anonymous referees for their constructive comments and suggestions. Also, the authors would like to express their hearty thanks to the Editorial Board Members for their cooperation. The third author would like to acknowledge the financial support provided by the DSTBT, Govt. of West Bengal, India vide Memo No. 30 (Sanc)/ST/P/S&T/16G-43/2017 Dated 12/06/2018.

References

Ahmadivala M, Mattrand C, Gayton N, Dumasa A, Yalamas T, Orcesi A (2019) Application of AK-SYS method for time-dependent reliability analysis. 24ème CongrèsFrançais de Mécanique Brest, 26 au 30 Août 2019

Ardakan M, Zahra M, Hamadani A, Elsayed EA (2017) Reliability optimization by considering time dependent reliability for components. Qual Reliab Eng Int 33(8):1641–1654

Bhattacharyee N, Kumar N, Mahato SK, Bhunia AK (2021) Development of a blended particle swarm optimization to optimize mission design life of a series-parallel reliable system with time dependent component reliabilities in imprecise environments. Soft Computing, Accepted

Bhunia AK, Sahoo L, Roy D (2010) Reliability stochastic optimization for a series system with interval component reliability via genetic algorithm. Appl Math Comput 216:929–939

Bhunia AK, Sahoo L (2011) Genetic algorithm based reliability optimization in interval environment. In: Nedjah N (ed) Innovative computing methods and their applications to engineering problems, SCI 357. Springer-Verlag, Berlin Heidelberg, pp 13–36

Chen DX (1977) Fuzzy reliability of the bridge circuit system. Syst Eng Theory Pract 11:109–112

Chern MS, Jan RH (1986) Reliability optimization problems with multiple constraints. IEEE Trans Reliab 35(4):431–436

Garg H (2015) Multi-objective optimization problem of system reliability under intuitionistic fuzzy set environment using cuckoo Search algorithm J. Intell Fuzzy Syst 29(4):1653–1669
Garg H, Rani M (2013) An approach for reliability analysis of industrial systems using PSO and IFS technique. ISA Trans Elsevier 52(6):701–710
Garg H, Rani M, Sharma SP (2014a) An approach for analyzing the reliability of industrial systems using soft-computing based technique. Expert Syst Appl 41(2):489–501
Garg H, Rani M, Sharma SP, Vishwakarma Y (2014b) Intuitionistic fuzzy optimization technique for solving multi-objective reliability optimization problems in interval environment. Expert Syst Appl 41(7):3157–3167
Garg H (2013) An approach for analyzing fuzzy system reliability using particle swarm optimization and intuitionistic fuzzy set theory. J Multi Val Logic Soft Comput 21(3–4):335–354
Garg H (2016) A novel approach for analyzing the reliability of series-parallel system using credibility theory and different types of intuitionistic fuzzy numbers. J Braz Soc Mech Sci Eng 38(3):1021–1035
Garg H (2017) Performance analysis of an industrial system using soft computing based hybridized technique. J Braz Soc Mech Sci Eng 39(4):1441–1451
González LCM, Rodríguez Borbón MIR, Valles-Rosales DJ, Valle AD, Rodriguez A (2015) Reliability model for electronic devices under time varying voltage. Qual Reliab Eng Int 32(4):1295–1306
Gupta R, Bhunia AK, Roy D (2009) A GA based penalty function technique for solving constrained redundancy allocation problem of series system with interval valued reliabilities of components. J Comput Appl Math 232(2):275–284
Hamadani AZ, Khorshidi HA (2013) System reliability optimization using time value of money. Int J Adv Manuf Technol 66(14):97–106
Hu and Mahadevan (2015) Time-dependent system reliability analysis using random field discretization. J mech design 137(10). https://doi.org/10.1115/1.4031337
Huang H (1996) Fuzzy multi-objective optimization decision-making of reliability of series system. Microele Reliab 37(3):447–449
Hwang CL, Tillman FA, Kuo W (1979) Reliability optimization by generalized Lagrangian-function based and reduced-gradient methods. IEEE Trans Reliab 28:316–319
Jamkhaneh E (2017) System reliability using generalized intuitionistic fuzzy exponential lifetime distribution. Int J Soft Comput Eng (IJSCE) 7:2231–2307
Kennedy J, Eberhart R (1995) Particle swarm optimization. In Proceedings of ICNN'95-International Conference on Neural Networks 4:1942–1948
Khalili-Damghani K, Abtahi AR, Tavana M (2013) A new multi objective particle swarm optimization method for solving reliability redundancy allocation problems. Reliab Eng Syst Saf 111:58–75
Kuo W, Prasad V, Tillman FA, Hwang CL (2001) Optimal reliability design fundamentals and applications. Cambridge University Press, Cambridge LINGO User Guide (2013) Lindo Systems Inc, Chicago, IL
Mahapatra GS, Roy TK (2012) Reliability evaluation of bridge system with fuzzy reliability of components using interval nonlinear programming. Electron J Appl Stat Anal 5(2):151–163
Mahapatra GS, Roy TK (2006) Fuzzy multi-objective mathematical programming on reliability optimization model. Appl Math Comput 174(1):643–659
Mahapatra GS, Roy TK (2009) Reliability evaluation using triangular intuitionistic fuzzy numbers arithmetic operations. World Acad Sci Eng Technol 50:574–581
Mahapatra GS, Roy TK (2011) Optimal redundancy allocation in series-parallel system using generalized fuzzy number. Tamsui Oxford J Inf Math Sci 27(1):1–20
Mahato SK, Bhattacharyee N, Paramanik R (2020) Fuzzy reliability redundancy optimization with signed distance method for defuzzification using genetic algorithm. Int J Oper Res (IJOR) 37(3):307–323
Mahato SK, Bhunia AK (2016) Reliability optimization in fuzzy and interval environments. Lap Lambert Academic Publishing (Book)

Mahato SK, Sahoo L, Bhunia AK (2012) Reliability-redundancy optimization problem with interval valued reliabilities of components via genetic algorithm. J Inf Comput Sci (UK) 7(4):284–295
Mahato SK, Sahoo L, Bhunia AK (2013) Effects of defuzzification methods in redundancy allocation problem with fuzzy valued reliabilities via genetic algorithm. Int J Inf Comput Sci 2(6):106–115
Misra K (1986) On optimal reliability design: a review. Syst Sci 12(4):5–30
Misra B, Sharma U (1991) An efficient algorithm to solve integer-programming problems arising in system-reliability design. IEEE Trans Reliab 40(1):81–91
Mori Y, Ellingwood BR (1993) Time-dependent system reliability analysis by adaptive importance sampling, vol 12, issue 1, pp 59–73
Mourelatos ZP, Majcher M, Pandey V, Baseski I (2015) Time-dependent reliability analysis using the total probability theorem. ASME J Mech Des 137(3). https://doi.org/10.1115/1.4029326
Nakagawa Y, Miyazaki S (1981) Surrogate constraints algorithm for reliability optimization problems with two constraints. IEEE Trans Reliab 30(2):181–184
Nakagawa Y, Nakashima K (1977) A heuristic method for determining optimal reliability allocation. IEEE Trans Reliab 26(3):156–161
Park K (1987) Fuzzy apportionment of system reliability. IEEE Trans Reliab 36(1):129–132
Sahoo L, Bhunia AK, Kapur PK (2012) Genetic algorithm based multi-objective reliability optimization in interval environment. Comput Ind Eng 62(1):152–160
Sahoo L, Bhunia AK, Mahato SK (2014) Optimization of system reliability for series system with fuzzy component reliabilities by genetic algorithm J. Uncertain Syst 8(2):136–148
Sahoo L, Bhunia AK, Roy D (2010) A genetic algorithm based reliability redundancy optimization for interval valued reliabilities of components. J Appl Quant Methods 5(2):270–287
Sahoo L, Bhunia AK, Roy D (2013) Reliability optimization with high and low level redundancies in interval environment via genetic algorithm. Int J Syst Assur Eng Manage. https://doi.org/10.1007/s13198-013-0199-9
Sun XL, Li D (2002) Optimization condition and branch and bound algorithm for constrained redundancy optimization in series system. Optim Eng 3(1):53–65
Sung CS, Cho YK (2000) Reliability optimization of a series system with multiple choice and budget constraints. Eur J Oper Res 127(1):159–171
Tillman FA, Hwang CL, Kuo W (1980) System effectiveness models: an annotated bibliography. IEEE Trans Reliab 29(4):295–304
Wang L, Wang X, Wang R, Chen X (2015) Time-dependent reliability modeling and analysis method for mechanics based on convex process, mathematical problems in engineering, vol 2015, Article ID 914893, p 16. http://dx.doi.org/https://doi.org/10.1155/2015/914893
Zafar T, Wang Z (2020) Time-dependent reliability prediction using transfer learning. Struct Multidiscip Optim 62:147–158. https://doi.org/10.1007/s00158-019-02475-5
Zhu Z (2016) Efficient time-dependent system reliability analysis Doctoral Dissertations, 2552. https://scholarsmine.mst.edu/doctoral_dissertations/2552