IJRRS
International Journal of Reliability, Risk Saftey: Theory and Application

Online ISSN: 2676-3346

Vol. 5/ Issue 2/ 2022/ pp. 41-48
DOI: 10.30699/IJRRS.5.2.5
Received: 2022.10.29, Accepted: 2023.02.01

Original Research Article

Optimal Multi-Level Redundancy Allocation Using a New Modified Bat Algorithm

Mohammad Ali Farsi[1*]

1. Aerospace Research Institute, Ministry of Science and Research, Tehran, Iran

*** Farsi@ari.ac.ir**

Abstract

One of the most important steps to design an engineering system is reliability allocation. Often, redundancy is used to achieve a highly reliable system. The redundancy allocation problem (RAP) is increasingly becoming an important tool in the initial stages of or prior to the plan, design, and control of systems. The multi-level redundancy allocation problem (MLRAP) is an extension of the traditional RAP such that all available items for redundancy (system, module, and component) can be simultaneously chosen. Although RAP has been considered by several researchers, MLRAP attracts only a little attention. Ordinarily, reliability uncertainty is ignored too. In this paper, this subject is studied and a new method to solve MLRAP is developed. The total cost is considered the most important constraint. A new meta-heuristic optimization algorithm, called Modified bat algorithm (MBA), to solve the constrained optimization problem (MLRAP) is proposed. This method is based on the Bat behavior to detect a prey. To demonstrate this method's capability, MLRAP for a system is described. The results are comprised with HGA, MA, and two-dimensional arrays encoding and a hybrid genetic algorithm (TDA-HGA). For this system, optimal results are the same as TDA-HGA and better than HGA and MA in all cases. Also, the reliability uncertainty and its influence on reliability allocation are studied. The optimal result is changed when uncertainty is considered. The proposed method is a simple and powerful tool to determine the optimal multi-level redundancy allocation and reliability uncertainty modeling.

Keywords: Modified bat algorithm; Reliability allocation; Redundancy allocation; Multi-level systems; Uncertainty.

1. Introduction

In real-world systems, performance is very important for customers to buy and use an item. The overall performance of an item (component, device, product, subsystem, or system) is dependent on the implementation of various programs that ultimately improve the performance of the item. The performance of an item can be described by four elements: Capability, efficiency, reliability, and availability. Reliability has a key role to achieve an item with high performance. Reliability is an operation-related issue and is influenced by the item's potential to remain operational. In other words; reliability is the ability of an item (a product or a system) to operate under desired operating conditions for a designated period of time or number of cycles [1]. Therefore, the reliability should be determined and predicted in the initial steps of the design cycles of a new system. In this step, usually, reliability allocation is done to achieve a desired reliability value. The typical reliability allocation problem may be stated as the maximization of the system reliability subject to some budget constraint, or the minimization of the system cost subject to the attainment of some specified level of the system reliability.

Nowadays, the redundancy allocation problem (RAP) is increasingly becoming an important tool in the initial stages of or prior to achieving a desired reliability value and predefined performance. Industrial systems include several modules and subsystems that each subsystem is divided into several units and finally, several components (parts) are used to build a unit. The RAP is one of the most important reliability optimization problems with regard to improving the reliability of real-world systems in the design phase. The most optimal redundancy allocation methods try to solve a problem at the system level and other items are ignored. In real-world systems (for example, communication systems, computing systems, control systems, and critical power systems), the influence of subsystems and components on system reliability cannot be ignored. Therefore, all available items for redundancy (system, module, and component) should be considered. For example, in a computer, we can use a motherboard or a CPU as redundancy, which is better?

The multi-level redundancy allocation problem (MLRAP) is an extension of the traditional RAP such that

all available items for redundancy can be simultaneously chosen.

The RAP has proved to be an NP-hard problem. Thus, how to find effective approaches to it is a hot topic. There are many approaches proposed to cope with RAPs, including exact methods, max-min approaches, dynamic programming, heuristic methods, and meta-heuristic algorithms. Among them, meta-heuristic methods, especially the genetic algorithms (GAs), are widely, and successfully used due to their robustness, and strong ability of global search, even though sometimes they are time-consuming. Although RAP has been considered by several researchers [2-5], attention to MLRAP is less. From these reviews, it can be observed that, in spite of its importance in the real-world system, the MLRAP has rarely been investigated. When the reliability value includes uncertainty, the reliability allocation becomes more complex.

G. Levitin proposed an algorithm based on a GA framework with a universal generating function technique for the system survivability evaluation [6]. This method aims to solve multi-level protection cost minimization problems subject to survivability constraints. Later, Yun and Kim proposed a restricted multi-level redundancy allocation model and addressed a tri-level RAP using a customized GA [7]. However, the customized GA employs a fixed-length vector to represent the solution to the MLRAP, and thus redundancy is required to be allocated to only one unit in a direct line, which is defined as a set of units from the system-level unit down to one component unit. Though both introduced methods were designed to cope with MLRAPs, they were based on strong assumptions or rigid rules that do not accord with reality. Recently, Kumar et al. [4,8] re-formulated MLRAPs so that redundancy allocation can be carried out at any level of a multi-level system. In this work, a solution to an MLRAP was represented by a hierarchical structure, and a simple GA was adapted to tackle the MLRAP involved. He et al. [9] proposed two-dimensional array encoding and a hybrid genetic algorithm (TDA-HGA). Their method is very useful to determine the best structure of a system. Although these studies have been shown to be effective via empirical studies, their effectiveness on the MLRAP can still be improved significantly. Torrado et al. proposed a model to study redundancy mechanisms at multiple levels [10]. Chung applied PSO for the optimization of a MLRAP [11].

Coit D.W. (2004)[12] proposed a method to model reliability uncertainty and its influence on system reliability. This method has been implied to Series-Parallel systems. MLRAP with reliability uncertainty has not been considered by other researchers (according to the author's lecture review.).

According to the lecture review, it can be said that RAP has been deeply studied by researchers. But; MLRAP is a new aspect of RAP. Since, to choose a redundant item in a system, several strategies may be selected. For example, the Processor core is the lowest-level item and the computer case is the top-level item. A company may use two cases as redundancy, but a user may use two motherboards, another user can apply two CPUs, and another user exploits four processor cores, etc. which strategy is good? The MLRAP should be considered to answer this question.

In this paper, a new heuristic method (hybrid Bat Algorithm) to solve MLRAP is proposed. This method uses the Bat movement principle to find prey. This method is simple and very fast to find the best result. This method is described in the next section. Also, reliability uncertainty is modeled to determine the optimal redundancy allocation.

2. Bat Algorithm

Bats are fascinating animals. Microbats are a type of bats that they use a type of sonar, called echolocation, to detect prey, avoid obstacles, and locate their roosting crevices in the dark. If we idealize some of the echolocation characteristics of microbats, we can develop various bat-inspired algorithms or Bat Algorithms (BA). In echolocation, each pulse only lasts a few thousandths of a second (up to about 8–10 ms). However, it has a constant frequency which is usually in the region of 25–150 kHz corresponding to the wavelengths of 2–14 mm. In BA, the echolocation characteristics of microbats can be idealized as the following rules [13,14]:

1. All bats use echolocation to sense distance, and they also ‘‘know’’ the difference between food/prey and background barriers in some magical way;
2. Bats randomly fly with velocity vi at position xi with a fixed frequency fmin, varying wavelength, and loudness A0 to search for prey. They can automatically adjust the frequency (or wavelength) of their emitted pulses and adjust the rate of pulse emission r ϵ [0,1], depending on the proximity of their target;
3. Although the loudness can vary in many ways, it is assumed that the loudness varies from a large (positive) A0 to a minimum constant value Amin. A0 in this research (optimal reliability allocation) is 1 and Amin is zero. The basic steps of BA can be summarized as the pseudo-code shown in Fig. 1.

For each Bat (i), its position x_i and velocity v_i in a N-dimensional search space should be specified. x_i and v_i should be subsequently updated during the iterations, since bat should be moved to target (prey). The new solutions x_i^t and velocities v_i^t at time step t are determined by:

$$f_i = f_{min} + (f_{max-} - f_{min})\beta \quad (1)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x^*)f_i \quad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

Where β in the range of [0,1] is a random vector drawn from a uniform distribution. Here, x* is the current

global best location (solution), which is located after comparing all the solutions among all the n bats. As the product $(x_i^{t-1} - x^*)f_i$ is the velocity increment. For implementation, f_{min} = 0 and f_{max} = 1.0 are used in this research. Initially, each bat is randomly assigned a frequency that is drawn uniformly from [0, 1].

Bat Algorithm

Objective function f(x), x=(x_1,x_2,…,x_n)
Initialize the bat population x_i and v_i
Define plus frequency f_i at x_i
Initialize pulse rates r_i and the loudness A_i
While (t<max number of iterations)
Generate new solution by adjusting frequency,
And updating velocities and locations/solutions (eq. 2 to 4)
If rand > r_i
Select a solution among the best solutions randomly
Generate a local solution around the selected best solution by a local random walk
End if
If (rand<A_i and f(x_i)<f(x*))
Accept the new solutions
Increase r_i and decrease A_i
End if
Rank the bats at each iteration and find their current best x*
End while
Post process results and visualization

Figure1. Pseudo code of BA

For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using a local random walk:

$$x_{new} = x_{old} + \varepsilon A^t \quad (4)$$

Where, the random number ε is drawn from [-1, 1], while A^t is the average loudness of all the bats at this time step.

The update of the velocities and positions of bats has some similarities to the procedure in the standard particle swarm optimization [15] as f_i essentially controls the pace and range of the movement of the swarming particles. To a degree, BA can be considered as a balanced combination of the standard particle swarm optimization and the intensive local search controlled by the loudness and pulse rate.

Furthermore, the loudness A_i and the rate r_i of pulse emission have to be updated accordingly the iterations proceed. Once a bat has found its prey, the loudness usually decreases and the rate of pulse emission increases. The loudness can be chosen as any value of convenience. In this paper, A_0 = 1 and A_{min} = 0 are used. Assuming A_{min} = 0 means that a bat has just found the prey and temporarily stop emitting any sound, we have:

$$A_i^{t+1} = \delta A_i^t, \quad r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \quad (5)$$

Where δ and γ are constants. In fact, δ is similar to the cooling factor of a cooling schedule in the simulated annealing [16]. For any $\delta > 0$ and $\gamma < 1$:

$$A_i^t \to 0, \quad r_i^t \to r_i^0, \quad as \quad t \to \infty \quad (6)$$

According to our study and simulation results, $\delta = \gamma = 06 - 0.9$ can be used for reliability allocation problem modeling.

The structure of BA has been presented above; according to the performance of BA; this algorithm is mentioned by several researchers and has been developed very fast. Mirjalili et al. developed a binary bat algorithm to perform global optimization. They compared their algorithm with binary PSO and GA using several examples and shown this method has superior performance [17].

Gandomi and yang introduced chaos into BA so as to increase its global search mobility for robust global optimization [18]. Ali used BA for the optimal design of Power System Stabilizers in a multi-machine environment [19]. A novel hybrid bat algorithm as Bat algorithm with differential evolution is proposed to enhance the performance of the basic BA by Meng et al. [20]. A hybrid metaheuristic HS/BA method for the optimization problems is presented by wang and Guo [21]. They improved the BA by combining the original harmony search (HS) algorithm and evaluating the HS/BA on multimodal numerical optimization problems. Yılmaz and Küçüksille worked on local and global search characteristics of BA enhanced through three different methods to modify BA for solving optimization problems [22].

Several researchers worked on BA a tried to increase the capability of this system and applied BA to train neural networks [23,24], solve scheduling work flow problems [25], and practical reserve-constrained dynamic environmental/economic dispatch problems [26]. Asharafi and Hassan used to optimize the reliability of a complex system [27]. Guerraiche et al. proposed a hybrid algorithm based on BAT algorithm and Generalized Evolutionary Walk to Reliability Optimization of Wind Farm Power Systems [28].

2.1 Modified Bat Algorithm

An original bat algorithm has been proposed for continuous variables. In this paper, we develop this algorithm for the discrete variables. Therefore, this algorithm should be modified. In the proposed algorithm, random walking and random fly are used and the next location (solution) is determined according to predefined data (or available values). In this condition, the closest state is selected as the next location for each bat. For example, a bat is characterized with A, B, and C parameters, in the random fly; all these parameters are calculated using equations 3 and 4. Then, distances between this new value and predefined values are compared. The closest value to the determined value is selected as the next value. After this step, this new location is checked by problem constraints such as cost. If this location can satisfy the constraint, this value is accepted. To improve algorithm capability to find the best solution, in each iteration, some bats are motivated.

This motivation increases convergence speed to find the best solution and increases the search field. Therefore, some of the bat position and location is changed randomly. We call it motivation. In this process, some bats (10% of bats) are selected according to their position (out of the best zone bats). Then, their positions are changed randomly. Thus, the search process is modified. The proposed method in this paper is called the Modified bat algorithm (MBA). This algorithm is applied via a developed Visual Basic.Net Program.

Problem modeling and handling of constraints to MLRAP are explained in the following sections.

3. Problem Describtions

In this section, we will first present the background description of MLRAP and then provide the formulation of this problem. To avoid confusion, the differences between the three most commonly used words in this paper are first presented below:

- Component: the elementary part of a system, at the lowest level.
- Unit: any part of a system either refers to the system, subsystems, or components.

Also, the MLRAP investigated in this paper is formulated based on multi-level serial systems with the following assumptions.

- *Assumption 1*: The redundancy can be allocated to the units on any level.
- *Assumption 2*: The quality (reliability, cost) of each component is predefined. If one unit is not a component, its reliability, and cost are calculated based on its child units.

3.1 Multi-level serial system

A multilevel serial system is defined as a hierarchical system with the entire system at the topmost level, the subsystems at lower levels, and the components at the lowest level. The system, subsystems, and components are all referred to as units. Child units of one unit refer to a fixed number of serial units at its immediately lower level while parent unit refer to the unit at the immediately higher level. Fig. 2 shows an example of multilevel serial systems. As Fig. 2 shows, U1 is a system unit with three serial child units U11; U12 and U13. U11 is the parent unit of three units: U111; U112 and U113. This structure applies to all of the units except for the component units, which have no child units.

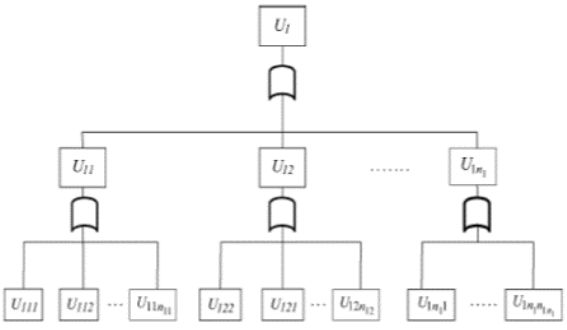

Figure 2. A general Multi level series system

3.2 Multi-level Series Redundancy Allocation Model

Redundancy could be configured to any unit from system level to component level. Once the redundancy is allocated to units at the parent level, there will be multiple child units at lower levels. Fig. 3 illustrates an example of a multilevel redundancy allocation model of a bi-level serial system. Fig. 3 consists of one system unit and two component units. The redundancy of unit U_1 is 2, so there are two groups of child units below U_1^1 and U_1^2 respectively. Under parent unit U_1^1 and U_1^2, the redundancy for unit U_{11} and U_{12} is (3,1) and (2,2) respectively. Example in Fig. 3 indicates that it might be difficult to represent the redundancy information in one multilevel system, because the overall number of the replica of one unit at the child level is determined by the redundancy of its parent unit and the redundancy of itself. For example, the redundancy of unit U_1 is 2, so there are two groups of child units U_{11}. The redundancy of child units U_{11} belonging to a different parent unit U_1^1 and U_1^2 is 3 and 2 respectively. So the overall number of replica of U_{11} is 3 + 2 = 5.

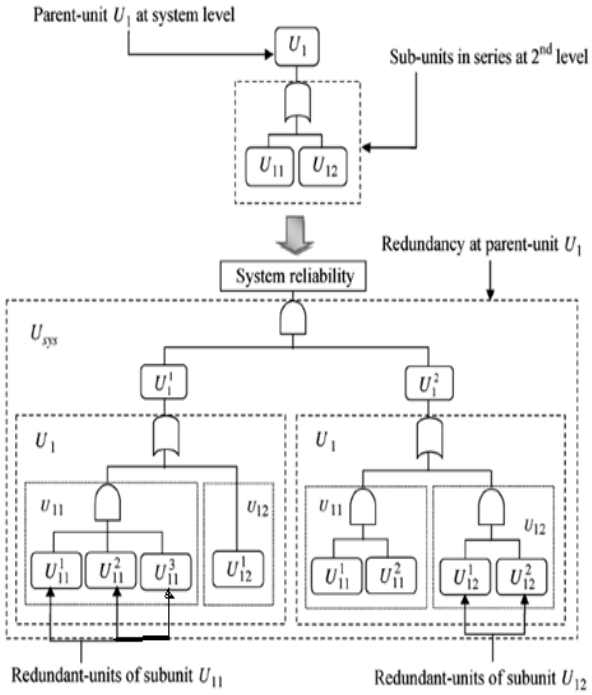

Figure 3. An example of redundancy allocation in bi-level series configuration

3.3 Redundancy Allocation Optimization Problem

In a multi-level serial system after redundancy allocation, the reliability of the higher level units could be obtained from the reliability of all its child units including all replicas. In general, a given unit U_i in the multilevel system has n_i subunits, $U_{i1}; U_{i2}; \ldots; U_{in_i}$, which must be connected either in series or in parallel. When x_i is the number of U_i redundant units, there are $n_i x_i$ sub-units in the level below U_i. A unit in the j^{th} redundant unit of the

m^{th} sub-unit of U_i is denoted $U_{i,m}^{j}$. Thus, the reliability R_i of unit U_i for multilevel series and parallel configurations can be calculated using the following equations:

$$R_i = \prod_m^{n_i}\left[1 - \prod_j^{x_i}(1 - R_{i,m}^{j})\right] \quad (7)$$

$$R_i = 1 - \prod_m^{n_i}[\prod_j^{x_i}(1 - R_{i,m}^{j})] \quad (8)$$

Where $R_{i,m}^{j}$ are reliability values of the sub-unit $U_{i,m}^{j}$. Each $R_{i,m}^{j}$ value is calculated using the above equation at the level immediately below the unit, and these calculations are recursively iterated to the level just above the very lowest hierarchical level.

Using Eq. (7 and 8), the reliability of the system could be calculated. For example, the reliability of the system shown in Fig. 2 is calculated as Eq. (9).

$$R = 1 - \left(1 - \left[\left(1 - \prod_{j=1}^{3}(1 - R_{11}^{j})\right)(R_{12}^{1})\right]\right) \times \left(1\left[\left(1 - \prod_{j=1}^{2}(1 - R_{11}^{j})\right)\left(1 - \prod_{j=1}^{2}(1 - R_{12}^{j})\right)\right]\right) \quad (9)$$

In a multi-level system, the cost can be calculated as the sum of its child units and some additional costs. Wang et al. [29] suggested following Eq. for the system cost calculation. This equation has been used by the most of research to cost modeling, we also use this equation.

$$C_i = \sum_{m=1}^{n_i}\left(\prod_{j=1}^{x_{im}} C_{im}^{j} + (\lambda_{im})^{x_{im}}\right) \quad (10)$$

As shown by Eq. (10), the cost C_i of unit U_i may be calculated from $C_{i,m}^{j}$, the cost of j^{th} replica of m^{th} child unit and λ_{im}, the additional cost of m^{th} child unit when configuring redundancy. x_{im} also denotes the number of redundant units in the m^{th} serial child unit and n_i denotes the number of child units under U_i. Wang et al. [15] specified that to reflect the hierarchical structure, the additional cost is only introduced in the component level. In our model, we do not specify it but allow the different value of k_{im} for each unit. We only need to set $\lambda_{im} = 0$ for all the units except for the component units to meet the assumptions made by Wang [25].

For the system illustrated in fig 2, using this equation, the system cost is calculated as Eq.11.

$$C_{sys} = \left[\left(\sum_{j=1}^{3} C_{11}^{j} + (\lambda_{11})^{3}\right) + (C_{12} + \lambda_{12})\right] + \left[\left(\sum_{j=1}^{2} C_{11}^{j} + (\lambda_{11})^{2}\right) + \left(\sum_{j=1}^{2} C_{12}^{j} + (\lambda_{12})^{2}\right)\right] + (\lambda_1)^2 \quad (11)$$

Reliability uncertainty is very important to design a system. Designers use confidence to define reliability value since they have a limitation in data and reliability tests. For example, when 15 components are tested and all components succeed in this test, reliability is 0.86 with 90% confidence. Uncertainty of system reliability depends on the reliability uncertainty of components. Thus, uncertainty should be modeled and tried to decrease. In an optimal system, the reliability is highest and uncertainty is lowest. Usually, to define reliability uncertainty, variance is used. To determine system variance from components several methods have been proposed by coit [12], Guo H. et al. [30], and James C. Spall [31]. These methods are used for series-parallel systems, we develop the Guo method and use it for multi-level systems and we assume redundancy items are parallel. For variance modeling, Eq.12 (For serial components) and 13 (For parallel components) are implied.

$$var(X) = \prod_{i=1}^{k}[R_i^2 + var(x_i)] - \prod_{i=1}^{k} R_i^2 \quad (12)$$

$$var(X) = \prod_{i=1}^{k}[(1 - R_i)^2 + var(x_i)] - \prod_{i=1}^{k}(1 - R_i)^2 \quad (13)$$

Where, k is number of components in system and $var(x_i)$ is component variance.

MLRAP is an NP-hard problem. To find the optimal state, we use a hybrid BA. In real-world problems, several constraints such as cost, volume, mass, etc impress the optimal solution. Cost is an important constraint and often, its influence is more than other constraints. Thus, the typical MLRAP may be stated as the maximization of system reliability subject to some cost constraint. In this paper, MLRP is formulated as shown by Eq. 7 and 8.

3.4 Fitness function

MLRAP is formulated as a constrained optimization problem, thus, violations of the constraints must be considered together with the value of objective functions when evaluating a solution. The fitness function for evaluating the quality of a solution during the search process of our method is defined by Eq. 14 and 15. When only reliability and cost should be determined, eq. 14 is implied. In this equation, a penalty function is used (Eq. 16) and we try to find the maximum of the fitness function. If reliability uncertainty is considered, Eq. 15 is proposed for fitness function modeling.

$$ff(b) = R \times \frac{c_0}{cost(b)} - pv \quad (14)$$

$$ff(b) = R \times \frac{c_0}{\sqrt{var(b) \times cost(b)}} - pv \quad (15)$$

$$pv = 100 * (1 - \frac{c_0}{cost(b)})^2 \quad (16)$$

Where; c_0 is cost constraint value, R is reliability, cost (b) is system cost and var(b) is system variance.

4. Case Study

In this section, the performance of our MBA is evaluated on an MLRAP example. The results are compared with MA[29], HGA[4] and TDA-HGA[9], which are the most recent algorithms in the literature. The structures of the multilevel serial systems are illustrated in Fig. 4 This system comprises three levels.

The MLRAP for this system has been used by He et al.[9], Wang et al.[29], and Kumar et al.[4] to evaluate the performance of TDA-HGA, MA, and HGA. To make a fair comparison between other algorithms and MBA, the control parameters set for MA [29] were also used in experiments for MBA.

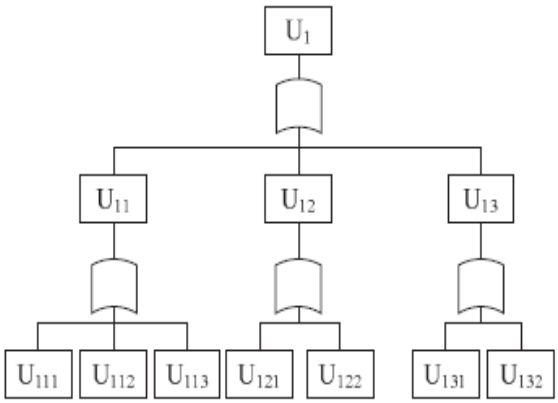

Figure 4. Case study

Table1. Input parameters

Unit	Reliability	Cost	λ	Confidence%
U_{111}	0.9	5	3	75
U_{112}	0.95	6	4	85
U_{113}	0.85	5	4	93
U_{121}	0.85	7	4	95
U_{122}	0.90	6	4	95
U_{131}	0.9	8	3	90
U_{132}	0.8	7	4	85

Specifically, the population size was set to 100, the maximum generation was 500, the crossover rate was set to 0.8, and the mutation rate was set to 0.1. In our algorithm, the local search rate was set to 0.2. For this system, the redundancy that can be allocated to a single unit was set between 1 and 5. The input parameters for the units in the two systems are shown in Table 1.

In this experiment, we aim to evaluate the convergence behavior of our MBA in a case study. The same group of input parameters was used by Wang et al.[17] was also used here. The cost constraint value was set to 150, and the parameters of the components of the system were given in Table 1. The best solution obtained in each generation was recorded, and the reliability of the corresponding system was calculated. The convergence curves of the MBA show that this method is very fast to find the best result. The detailed comparison of MBA and TDA-HGA, MA, and HGA will be given in the next sub-sections. Figure 5 shows the convergence curve of MBA on this problem. When the cost is 150, the proposed algorithm can find the best solution after 12 iterations and find the best solution for cost=150 after 6 iterations. Figure 6 shows an optimal structure for this example with a cost constraint value equal to 150 and 200. These cases have been run 10 times and in all runs, the optimal location is similar. In other words, for these cases, the variance is zero.

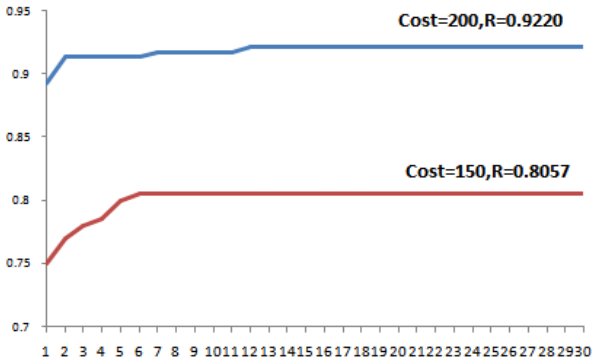

Figure 5. Convergence curve of MBA for two cases (30 Iterations)

4.1 Performance over Different Constraint Values

The previous experiment only provides a case study on which the MBA outperformed, and further comparisons between the four methods subject to different cost constraint values were carried out in this experiment. Seven cost constraint values were sampled between the intervals 150–340. All the system parameters were kept the same as in the previous sub-section

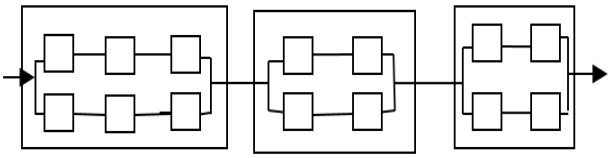

Figure 6. Optimal structure for system with cost constraint=150

For each cost constraint value, MBA, TDA-HGA, HGA, and MA were carried out 10 times. Table 2 summarizes the results of this experiment including the best reliability and cost obtained. For each cost constraint value, the reliability that is significantly better than the others is highlighted in boldface. From Table 2, we can observe that the proposed MBA outperformed the original MA and HGA in all the cases, in terms of both the quality of the best solutions and overall performance. In particular, the best solutions found by our MBA are always better than those found by MA and HGA. MBA performed similarly to TDA-HGA on best reliability in all of the test cases. If the convergence of these methods is compared together, the proposed algorithm is very fast; for example in cost equal to 300, the best solution is found at the 40th iteration and it is very faster than other methods (Table 3).

TDA-HGA is a method that it is improved in several versions, but our algorithm is the first version of MBA, therefore, we hope, the capability of this method is improved by developing the proposed algorithm next, especially, since we focus on the local search method in the next work.

Table2. MLRAP Results

Cost Con.	Best solutions							
	MA		HGA		TDA-HGA		MBA	
	Reli	cost	Reli	cost	Reli	cost	Reli	cost
150	0.8005	141	0.764	148	0.8057	140	0.8057	140
180	0.8781	179	0.8617	173	0.8826	179	0.8826	179
200	0.9032	198	0.8899	198	0.9220	200	0.9220	200
220	0.9371	220	0.9395	213	0.9456	210	0.9456	210
250	0.9628	249	0.947	250	0.9689	249	0.9689	249
280	0.9729	278	0.9647	279	0.9853	280	0.9853	280
300	0.9849	299	0.9730	292	0.9884	299	0.9884	299
340	0.993	338	0.9835	337	0.9943	340	0.9943	340

Table3. Comparison between MA, IMA, TDA-HGA and MBA convergence to find the best solution

Algorithm	MA[9]	IMA[9]	TDA-HGA[9]	MBA
Iterations No.	40	70	44	39
The best solution	0.9489	0.9882	0.9884	0.9884

4.2 Redundancy Allocation with Confidence Level

For the study on reliability uncertainty influence on redundancy allocation, system variance is modeled using eq. 12 and 13 and the fitness function is defined as Eq.15. This problem is solved again. Table 4 shows the optimal result for the system when uncertainty is ignored and it is considered.

Table 4. Redundancy allocation with reliability uncertainty

Cost constraint value	Best solution whit out variance			Best solution whit variance		
	Reliability	cost	variance	Reliability	cost	variance
150	0.8057	140	0.0067	0.8005	141	0.0023
160	0.84095	159	0.00164	0.84095	159	0.00164
180	0.8826	179	0.00083	0.8781	179	0.0005

This result shows reliability uncertainty can impress the optimal solution for MLRAP. The presented method is the initial version of this problem and should be studied more in future work.

5. Conclusions

The RAP has been proven to be an NP-hard problem. In this paper, the RAP on multi-level systems is considered. Multi-level redundancy allocation problem has a key role in real-world systems design. A Modified Bat algorithm (MBA) is proposed to solve this problem. This algorithm is based on micro bat flies to find prey. This method is improved and modified in this paper to find the best solution for an MLRAP. The Fundamental of this method is described and the capability of this method is demonstrated using a practical problem. This problem has been solved using Genetic Algorithm and Memetic Algorithm, our study shows that the proposed method is very fast and powerful to solve MLRAP. The best solutions found by our MBA are always better than those found by MA and HGA. MBA performed similarly to TDA-HGA on best reliability in all of the test cases. Also, the influence of reliability uncertainty is studied too. The proposed method determines the best solution according to cost constraint value and reliability increasing with decreased system variance. This problem is very complex and NP-hard, but the proposed MBA can find the best solution via a simple and fast method. Therefore, we hope the capability of this method is improved by developing the proposed algorithm in the next, especially; We focus on the local search method in the next work.

6. References

[1] M. Modarres, M. Kaminskiy, V. Krivtsov, Reliability engineering and risk analysis: a practical guide, CRC Press, (2009).

[2] M. A. Farsi and B. I Jahromi, "Reliability allocation of a complex system by genetic algorithm method ",International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering - ICQR2MSE, china, Chengdu, ,pp.1046-1049, (2011)

[3] E. Zio, Podofillini L.," Integrated optimization of system design and spare parts allocation by means of multiobjective genetic algorithms and Monte Carlo simulation", Proc

IMechE Part O: Journal of Risk and Reliability , 221,1:97-84, (2007).
[4] R. Kumar, K. Izui, M. Yoshimura, S. Nishiwaki, "Multilevel redundancy allocation optimization using hierarchical genetic algorithm", IEEE Trans. Reliability, 57, 4: 650–661, (2008).
[5] L. Sahoo, A. hunia, & P. Kapur, (2012), "Genetic algorithm based multi-objective reliability optimization in interval environment", Computers and Industrial Engineering, 62:152–160.
[6] G. Levitin, "Optimal multilevel protection in serial-parallel systems", Reliability Engineering and System Safety, 81, 1: 93–102, (2003).
[7] W. Y. Yun and J. W. kim, "Multilevel redundancy optimization in series systems", Computers and Industrial Engineering, 46:337–346, (2004).
[8] R. Kumar, Izui, K., Y. Masataka, "Optimal multilevel redundancy allocation in series and series–parallel systems", Computers and Industrial Engineering, 57:169–180. (2009).
[9] Pan He, Kaigui Wu, Jie Xu, Junhao Wen, Zhuo Jiang "Multilevel redundancy allocation using two dimensional arrays encoding and hybrid genetic algorithm", computer and industrial engineering, 64:69-83, (2013).
[10] N. Torrado, A. Arriaza, J. Navarro, A study on multi-level redundancy allocation in coherent systems formed by modules, Reliability Engineering & System Safety, Volume 213, 2021, 107694, (2021).
[11] Han Chung , Particle Swarm Optimization for Redundancy Allocation of Multi-level System considering Alternative Units, *J Korean Soc Qual Manag.* 2019;47 (4): 701-711
[12] Coit D.W," System Optimization with Component Reliability Estimation Uncertainty: A Multi-Criteria Approach", IEEE Trans. Reliability, 53, 3:369-380. (2004).
[13] Yang Xin-She, "Bat algorithm for multi-objective optimization", International Journal of Bio-Inspired Computation, 3, 5:267-274, (2011).
[14] Yang Xin-She, and A. H. Gandomi(2012), "Bat algorithm: a novel approach for global engineering optimization", Engineering Computations, 29, 5:464-483.
[15] J. Kennedy, and Eberhart, R., "Particle swarm optimization", Proceeding of IEEE Int. Conference Neural Networks, Perth, Australia, pp.1942-1945, (1995).
[16] S. Kirkpatrick, , Gelatt, C. D., Vecchi, M. P., "Optimization by simulated annealing", Science, New Series, 220,4598:671-680, (1983).
[17] S. Mirjalili, , Mirjalili, S. M., & Yang, X. S. , "Binary bat algorithm', Neural Computing and Applications, 25(3-4): 663-681, (2014).
[18] A. H.Gandomi, , & X. S. Yang, , "Chaotic bat algorithm", Journal of Computational Science, 5(2): 224-232, (2014).
[19] E. S. Ali, "Optimization of power system stabilizers using BAT search algorithm", International Journal of Electrical Power & Energy Systems, 61: 683-690, (2014).
[20] X. Meng, X. Z. Gao, , & Liu, Y. "A Novel Hybrid Bat Algorithm with Differential Evolution Strategy for Constrained Optimization", International Journal of Hybrid Information Technology, 8(1): 383-396, (2015).
[21] G. Wang, & Guo, L.." A novel hybrid bat algorithm with harmony search for global numerical optimization", Journal of Applied Mathematics, (2013)
[22] S. Yılmaz, , & Küçüksille, E. U.."A new modification approach on bat algorithm for solving optimization problems", Applied Soft Computing, 28: 259-275, (2015).
[23] M. Tuba, Alihodzic, A., & Bacanin, N.. "Cuckoo Search and Bat Algorithm Applied to Training Feed-Forward Neural Networks" In Recent Advances in Swarm Intelligence and Evolutionary Computation (pp. 139-162). Springer International Publishing, (2015).
[24] K. Khan, & A Sahai, "A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context", International Journal of Intelligent Systems and Applications (IJISA), 4(7), 23 . (2012).
[25] S. Raghavan,; C. Marimuthu,; Sarwesh, P.; Chandrasekaran, K., "Bat algorithm for scheduling workflow applications in cloud," *Electronic Design, Computer Networks & Automated Verification (EDCAV), 2015 International Conference on* , Shillong ,India, pp.139,144, (2015).
[26] T. Niknam,; Azizipanah-Abarghooee, R.; Zare, M.; Bahmani-Firouzi, B., "Reserve Constrained Dynamic Environmental/Economic Dispatch: A New Multiobjective Self-Adaptive Learning Bat Algorithm," Systems Journal, IEEE , 7(4):763,776, (2013).
[27] Abd Alsharify, F. H.., & Hassan, Z. A. H. Optimization of complex system reliability: Bat algorithm based approach. *International Journal of Health Sciences*, *6*(S1), 14226–14232. https://doi.org/10.53730/ijhs.v6nS1.8637, (2022).
[28] Kh. Guerraiche, E. Chatelet, L. Dekhici, and A. Zeblah Reliability Optimization of Wind Farm Power Systems Using Bat Algorithm with Generalized Flight, International Journal of Reliability, Quality and Safety EngineeringVol. 28, No. 03, 2150019, (2021).
[29] Z. Wang, K. Tang, & Yao, X., "A memetic algorithm for multi-level redundancy allocation". IEEE Transaction on Reliability, 59:754–765, (2010).
[30] H. Guo, Jiang M., Wang W.(2014),"A Method for Reliability Allocation with Confidence Level", Reliability and Maintainability Symposium, Colorado, USA, DOI: 10.1109/RAMS.2014.6798447
[31] J. C. Spall," System Reliability Estimation and Confidence Regions from Subsystem and Full System Tests", American Control Conference, Hyatt Regency Riverfront, St. Louis, MO, USA; pp. 5067-5072, (2009).