

An Estimation-of-Distribution Algorithm Approach to Redundancy Allocation Problem for a High-Security System

Haydee Melo^{*a)} Non-member, Junzo Watada^{*} Member

(Manuscript received Aug. 19, 2013, revised Feb. 3, 2014)

Reliability is an issue that has recently captivated the attention of researchers. Its goal is to develop new techniques to design more reliable systems, which can operate without failing during operation. A result of this growth in technology is an increase in the complexity and susceptibility of more complex systems. The principal objective of redundancy allocation is to maximize the availability of a system while reducing the cost, volume or weight. This paper proposes an Estimation-of-Distribution Algorithm (EDA) approach as a new meta-heuristic method to solve a redundancy allocation problem (RAP) for a high security control system.

Keywords: reliability, availability, maintainability, redundancy, estimation of distribution algorithm, PLC

1. Introduction

In 1998, Blanchard defined availability as a “measure of the degree of a system to which extent the system is operable and committable at the start of mission when the mission is called for at an unknown random point in time”⁽⁵⁾. However, this definition was developed based on Barlow and Proschan’s definition, given in 1975 as “the probability that the system is operating at a specified time t ”⁽⁴⁾. In 1977, Lie, Hwang and Tillman made a survey of a systematic classification of availability. In 1998, Mi compared the results of availability measures considering inherent availability. In 1975, Barlow and Proshman considered availability in maintenance modeling for a replacement model, Fawzi and Hawkes considered an r -out-of- n system with spares and repairs in 1991, then in 1992, they applied availability for a series system with replacement and repair for imperfect repair models. In 1995 Murdock studied age replacement preventive maintenance models.

The redundancy of a system can make critical production services work continuously, even when some parts of the system fail. Failures bring not only production loss or economic loss but also cause severe or even lethal injuries to a society. New techniques and methods are required for reducing these failures and the expenses of the out-of-operation costs and repair costs to improve system performance. Redundancy enables us to increase the system’s reliability and extend its operation time. However, understanding the causes and influence of these failures is essential to optimize the system redundancy. As a result, the study of failure is becoming an increasingly important issue.

At the same time, reliability analysis always plays a pivotal

role in keeping the level of system safety high⁽²³⁾. A failure of a single component in the system will affect the performance of the system or even stop its operation. Additionally, locating these failures is a time consuming task. Currently, the majority of recent approaches have focused on maximizing the reliability to ensure the continuity of system performance. One of the most used approaches is the redundancy allocation. The principal objective of redundancy allocation is to provide the corresponding redundancy level or a number of redundant components for each subsystem. However, in designing a complex system, the system consists of subsystems and each subsystem is composed of several subsystems or components.

Various designs are also constrained by cost, volume, weight and so on⁽¹⁸⁾. These constraints further complicate improving system reliability. In particular, the redundancy problem is an NP-hard problem where it is difficult to produce a solution with heuristic methods due to the huge search space⁽¹⁹⁾.

A GA has been widely used for several research studies on redundancy allocation for finding the minimal cost configuration of a series-parallel and r -out-of- n systems under the constraints of reliability or availability. Although, GA has yielded good results in solving various NP hard problems, it does not guarantee global optimization and many modifications of the GA algorithm are necessary for each problem. GA depends on the selection of the genetic operators (such as selection, crossover, mutation, probabilities of crossover and mutation, population size and number of generations) and problem specific interactions among the genes sometimes results in premature convergence. To avoid the disruption of partial solutions the crossover recombination processes can be replaced by generating new solutions according to the probability distribution of all promising solutions in the previous generation. This new approach is called Estimation of distribution Algorithm (EDA).

The objective of this research is to propose an Estimation-of-Distribution Algorithm (EDA) approach for solving the

a) Correspondence to: Haydee Melo. E-mail: melo.haydee@asagi.waseda.jp

* Graduate School of Information, Production and System, Waseda University
2-7, Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka 808-0135, Japan

redundancy allocation problem (RAP) for a high-security-system, because this algorithm can yield an optimal solution for combinatorial optimization problems by predicting the movements of the populations in the search space and avoiding the several parameters associated with other evolutionary computation algorithms⁽⁸⁾⁽¹⁷⁾. The motivation of this research is not only to develop a new formulation for the RAP but also to create an optimization model from the perspective of availability. It can assist engineers during the design phase by creating a highly reliable and available system.

The paper is organized as follows: Section 2 gives a review of previous research. Section 3 describes the mathematical formulation for a parallel-series redundancy allocation model. In Section 4 the EDA concept is explained, and the mathematical formulation for estimating the optimal solutions is illustrated. In Section 5, a real application is presented. The results are compared with the ones obtained by PSO. Finally, the results are discussed in Section 6. The conclusions are summarized in Section 7.

2. Previous Research

In recent years almost all of the studies on highly reliable systems have focused only on reliability without considering availability or maintainability⁽²³⁾. Many approaches are used to try to solve the problem by determining the redundancy allocation with heuristics methods and meta-heuristics methods.

The RAP is an NP-hard problem. Many different mathematical approaches have been used, such as integer programming to find an optimum solution. However, these approaches limited the search space artificially or made the problem linear, causing a loss of information and possible better solutions. Sung *et al.*⁽²²⁾ take the redundancy optimization problem as a problem with multiple-choice and resource constraints and propose a Branch-and-Bound heuristic method. This method can narrow the solution space for solving a non-linear integer programming problem.

Rajendra *et al.*⁽¹⁹⁾ transform the problem into a mixed integer linear programming by involving constraints in the problem variables. Their results encourage the use of their LP-based heuristic method. However, this algorithm did not show any experimental results in large structures due to the difficulty of finding a true optimal allocation by complete enumeration.

David W. Coit *et al.* use a meta-heuristic approach to solve the RAP for a series-parallel system⁽²⁶⁾. They used an example a system with 14 subsystems and with constraints on cost and weight. The problem was formulated as an integer programming one with un-separable constraints based on the work by Fyffe *et al.*⁽¹¹⁾, who used the GA to find the optimal solutions instead of the Lagrangian multiplier method. Coit also presented a column generation approach, which involves the maximization of the system's reliability by selecting components and redundancy levels based on the formulation of a constrained master problem and the generation of better possible solutions through a set of sub-problems. This approach assumes that an objective function and constraints are separable in decision variables. To use column generation the problem is decomposed into two problems: a constrained master problem and sub-problems, where each

sub-problem has a unique integer solution vector. However, the column generation did not guarantee optimal solutions because of its approximate computations. Coit and Smith⁽¹⁰⁾ introduce a multiple-weighted-objective heuristic method that transforms the problem from a multiple-objective optimization problem into a different single-objective problem. This method obtained good solution, although it is not as good as the best for the meta-heuristic.

All the methods above address the RAP only from the perspective of reliability. Another approach is to take into account the availability of the system instead of the reliability. K. K. Govil⁽¹²⁾ introduce the calculation of maintainability and availability for series, parallel and k -out-of- n configurations. In this research a formulation for availability includes the calculation of the maintainability of a single device or subsystem. However, the calculation is based on mean-time-to-availability (MTTA) and mean-time-to-failure (MTTF), without calculating reliability of each component.

3. Problem Description

The RAP is an optimal configuration problem of a series-parallel system that consists of s subsystems in series, in which it is necessary to determine the optimum number of components for each subsystem arranged in parallel⁽²⁵⁾ as shown in Fig. 1.

In this paper we demonstrate a mixed integer programming approach whose goal is to optimize the number of components j in subsystem i so that the availability $A(\mathbf{x})$ of the system is maximized for the decision variable $\mathbf{x} = [x_1, x_2, \dots, x_s]$ under the constraints of total cost C and total weight W . In this formulation, $A(\mathbf{x})$ is a function of the component vector \mathbf{x} . As using the notations, shown Table 8, the problem can be rewritten as follows:

$$\max_{\mathbf{x}} A(\mathbf{x}) = \prod_{i=1}^S A_i(x_i) \dots \dots \dots (1)$$

subject to:

$$\sum_{i=1}^S \sum_{j=1}^{n_i} c_{ij} x_{ij} \leq C \dots \dots \dots (2)$$

$$\sum_{i=1}^S \sum_{j=1}^{n_i} w_{ij} x_{ij} \leq W \dots \dots \dots (3)$$

3.1 Relationship between Reliability, Availability and Maintainability

Availability is the probability that the system does not fail or undergo repair and can operate when the system is required for use⁽¹³⁾. Availability is defined in

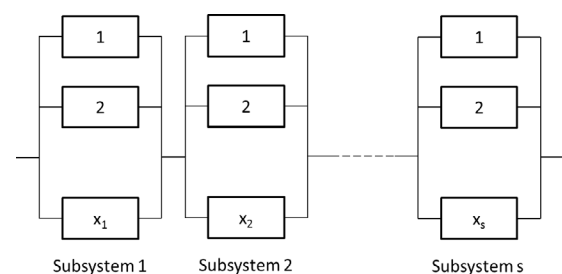


Fig. 1. Series-parallel system configuration

Table 1. Relationship between reliability, availability and maintainability

Reliability	Availability	Maintainability
Increase	Increase	Constant
Decrease	Decrease	Constant
Constant	Increase	Increase
Constant	Decrease	Decrease

equation (4) by using the mean time to failure (MTTF) and mean time to repair (MTTR) to express the reliability and maintainability, respectively. Therefore, different values in these measures can result in the same availability.

$$A(t) = \frac{MTTF}{MTTF + MTTR} \dots\dots\dots (4)$$

Any change in reliability or maintainability affects availability, see Table 1. The availability can be calculated by using an exponential law for the failure time and repair time. In the following ⁽¹⁰⁾⁽¹²⁾, the definition of the availability of the system consists of the reliability and the maintainability, both of which are probabilities.

$$A(t) = 1 - e^{\mu t}(1 - e^{-\lambda t}) \dots\dots\dots (5)$$

where μ is the maintenance rate, λ is the failure rate, t is the time period, $e^{\mu t}$ stands for the maintainability and $1 - e^{-\lambda t}$ for the reliability. This formulation of availability can also be expressed as:

$$A(t) = 1 - M(t)(1 - R(t)) \dots\dots\dots (6)$$

3.2 Series-Parallel Configuration Out of all the possible system designs for the RAP, let us consider the design of a series-parallel system. Maximizing Eq.(6) can determine the optimal number of components in a series-parallel system. Eq. (6) is rewritten in the following equation:

$$A(\mathbf{x}) = \prod_{i=1}^S \left[1 - \prod_{j=1}^n 1 - [m_{ij}(1 - r_{ij})] \right] \dots\dots\dots (7)$$

3.3 Multi-level Redundancy Allocation Optimization

The multilevel redundancy allocation model takes the hierarchical relationships among the different components into consideration. The system with the highest hierarchical level is named a master system, and the components in the lower level are named slave systems and slave components of their respective subsystems. Subsystems or components can be located in a intermediate or second level. Each system or subsystem can have a number of subordinate components or subsystems. These subordinate subsystems or components are referred to as slave subsystems or slave components, respectively ⁽¹⁵⁾.

The proposed redundancy allocation model is represented in the schematic diagram in Fig.2 for a generalized hierarchical redundancy allocation model to include the relationship of all components ⁽¹⁵⁾⁽²⁴⁾. Fig.2 explains the allocation of a multi-level series system and the differentiation between slave components and redundant components. S_1 is the master system with slave subsystems S_{11} , S_{12} and S_{13} , where S_{111} , S_{112} and S_{113} are considered slave components in the

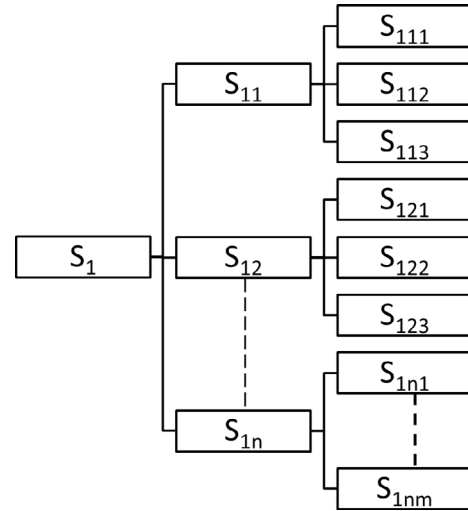


Fig. 2. Redundancy Hierarchical level

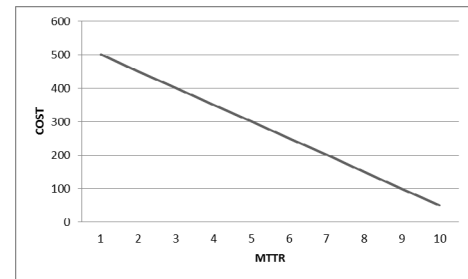


Fig. 3. MTTR and cost relationship

slave subsystem S_{11} . In this multilevel configuration each component can have redundant units, and slave subsystems, so there are S_{mn} slave subsystems in the lower levels.

Summing all the maintainability in all levels of redundancy, the total availability can be calculated by using:

$$A(x) = \prod_{i=1}^S \left[1 - \prod_{j=1}^n 1 - \{m_{ij}(1 - r_{ij})\}^{x_{ij}} \right] \dots\dots\dots (8)$$

3.4 Constraints Some actions are taken to avoid the stop or partial stop of the system due to an occurrence such as the repair of a component. In this case the repair actions would generate extra cost. Assuming a linear relationship between MTTR and the repair cost of components, as Fig.3 shows, a lower MTTR indicates a higher repair cost ⁽¹⁴⁾. Therefore, including the repair cost of components as a penalty for choosing a component with higher repair cost, the total cost of the system is re-formulated as follows:

$$\sum_{i=1}^S \sum_{j=1}^n (c_{ij} + c_{MTTR_{ij}})x_{ij} \dots\dots\dots (9)$$

4. EDA Optimization

The Estimation-of-distribution algorithm (EDA) is a branch of evolutionary computation that is considered to be the next generation of population-based algorithms ⁽²⁾. This algorithm incorporates knowledge of the previous studies into the search to learn about the problem structure. The EDA incorporates a learning technique by introducing a

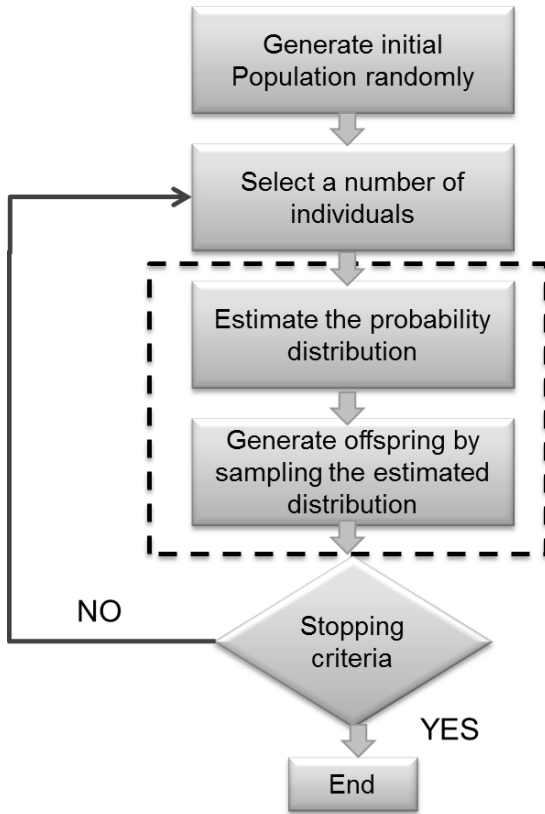


Fig. 4. Procedure of the Estimation-of-Distribution-algorithm

probabilistic distribution to the population. Unlike Genetic Algorithm (GA) that uses mutation or crossover for generating new offspring, EDA generate the new offspring according to a probabilistic distribution learned from a population of parents (learning technique). In this paper, we use an univariate marginal distribution algorithm (UMDA). The UMDA is suited to solve combinatorial problems in which the relationships among the problem variables are unclear. Let us assume that the probability distribution of each feature is marginal. No dependence between the problem variables is taken into account when learning the factorization. Thus, the n -dimensional joint probability distribution factorizes as a product of n -univariate and independent probability distributions.

4.1 Pseudo Code The EDA replaces the search operators with the estimation-of-distribution of the selected individuals that are sampled from this distribution as shown in Fig. 3. The objective of using the estimation-of-distribution is to avoid the use of arbitrary operators (mutation, crossover) and enhance the distribution of promising solutions. After each iteration, two actions are executed: one is to pursue a population with a probabilistic distribution. After the new individuals are evaluated, the best ones are selected and their distribution is estimated. The algorithm is written as follows:

- **Step 1:** Randomly generate M individuals (the initial population).
- **Step 2:** Repeat steps 3-5 for generations $l = 1, 2, \dots$ until the stopping criteria are met.
- **Step 3:** Select N individuals from D_l^i according to a random selection method.

- **Step 4:** Estimate the probability distribution $pl(\mathbf{x})$ of each individual among the selected individuals.
- **Step 5:** Sample M individuals (the new population) from $pl(\mathbf{x})$.

This procedure is repeated until the stopping criteria is satisfied.

4.1.1 Population The encoding of the problem uses several vectors to represent each level of the system, where x_{ij} is the designed variable denoting the redundancy for the i -th sub-system of the j -th redundancy unit, where j varies from 1 to k_i . Initially, the population is determined randomly. For each gene in the population, we consider the size of the subsystems in the corresponding hierarchical level of the problem as the size and range of the vector are defined as 1 as the minimum and n as the maximum of components in parallel. The population is scored using the objective function. From this ranked population, the objective function selects a subset of the best solutions. Then, the algorithm constructs a probabilistic model which attempts to estimate the probability distribution of the selected solutions.

$$pl(x) = \prod_{i=1}^n pl(x_i) \dots \dots \dots (10)$$

4.1.2 Objective Function The objective function or fitness function has been applied to transform the constrained problem into a non-constrained problem, by including a penalty function in the objective function to restrict infeasible solutions for any violation of the constraints. To guarantee the efficiency of the optimal solution for high constrained solutions, this function is used to calculate the fitness values of the solutions obtained in each generation.

$$evalf(\mathbf{x}) = A(\mathbf{x}) \times q(\mathbf{x}) \dots \dots \dots (11)$$

where, $A(\mathbf{x})$ and $q(\mathbf{x})$ are the system availability and penalty functions, respectively. We calculate the value of $q(\mathbf{x})$ for each individual. The penalty approach here adjusts the ratio of penalties adaptively at each generation to achieve a balance between the preservation of information, the selective pressure for infeasibility and the avoidance of excessive penalization.

4.1.3 Estimation In the EDA, a Gaussian distribution algorithm is usually used in the following for estimating the population distribution:

$$p_l(\mathbf{x}) = pl(\mathbf{x} | D_l^{Selected}) = \prod_{i=1}^n p_l(x_i) \dots \dots \dots (12)$$

However, the UMDA does not consider any dependencies between its variables⁽²¹⁾. The problems are assumed to have no interrelation among the variables. Hence, the n -dimensional joint probability distribution is factorized as a product of n -univariate and independent probability distribution⁽⁷⁾. The n -dimensional joint probability distribution $p_l(\mathbf{x})$ factorizes as a product of n - univariate and independent probability distributions $p_l(x_i)$, that is pl , where $x_i (i = 1, \dots, n)$ represents the individual of n genes and D_l denotes the population of individuals selected from the population in the l th generation. Similarly, D_l^N represents the population of the N individuals selected from D_l . This joint probability must be estimated every generation from one individual \mathbf{x} being

among the selected individuals. Therefore, each univariate marginal distribution is estimated from marginal variables as follows:

$$p_l(\mathbf{x}) = \frac{\sum_{j=1}^N \delta_i(X_i = x_i | D_l^{Selected})}{N} \dots \dots \dots (13)$$

where $\delta_i(X_i = x_i | D_l^{Selected}) = 1$ if j^{th} of the $D_l^{Selected}$ i^{th} gene is $X_i = x_i$ and 0 otherwise.

4.1.4 Sampling Function Once the probabilistic model has been estimated from the selected individuals the model is sampled to generate new individuals (new solutions). In this research the new population is sampled from a univariate Gaussian model. The objective is to sample new individuals that concentrate around the optimum solution.

5. High Security Facility System

This section is dedicated to the optimization of a real high security system. The system is located in a city in Northern Mexico. This control system is designated to operate the opening and closing of doors and controls of the alarm and communication systems. The system is principally composed of a control system, sensors, magnetic actuators, human-machine interface (HMI) and communication network between the different areas. The facility is a three-story building, each floor is divided into 9 sections (V_0 to V_9) as depicted in Fig. 5. In each sections there is a security room where several zones are monitored and by the security experts. In sections V_0 and V_1 , the security room that access to the control room is located. In section V_2 is the access to the hallway to the control system. The section V_3 and section V_9 control and monitor the access to the machine room and storage room respectively. In section V_4 , the security room that controls and monitors the access to the hallway for accessing for the following sections V_5 , V_6 , V_7 and V_8 that controls and monitors the adjacent zones (Zone 1 to Zone 9).

The master system is located at V_0 where it is responsible for access to the machine room, the general storage room, and their respective secure rooms, monitoring and alarm recordings and the information to PCs. The slave systems are located in the different sections for controlling the adjacent

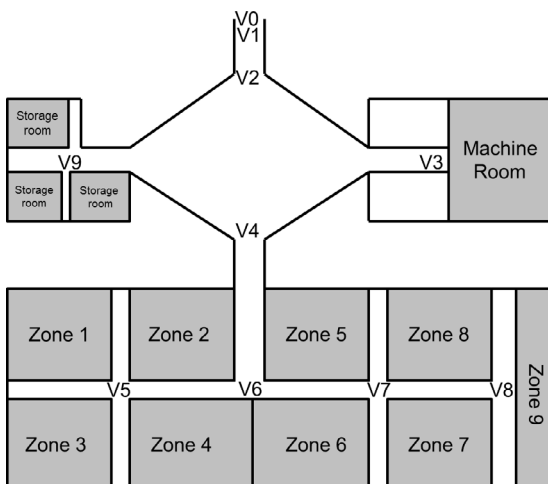


Fig. 5. High-safety system layout

zones.

The control system has an hierarchy structure that is divided in 3 levels as depicted in Fig. 6, where each level is responsible for the control of a group of sections. The Functional structure of the high security control system is depicted in Fig. 7.

The system has a PC and a master PLC (Programmable Logic Controller) with 7 identical slaves PLC's that each control another slave PLC. Each slave PLC is a control subsystem with different components: CS1-PA207R (power supply), CS1-CPU67H (CPU), S8VS-090 (power supply), CS1-IC102D (analog input model), CS1-ID261 (digital input) and CS1-OD261(digital output); see Table 2. Each component is critical and indispensable to each subsystem, and a malfunction of any component may interrupt the data communication and hence cause a loss of control of the subsection.

- (1) **Slave system 1 (containing CPU1).** This slave subsystem controls sections V_0 which and V_1 , where the control room is located, coordinates the communication between the others CPU and the control room, and controls the principal access to the facility.
- (2) **Slave system 2 (containing CPU2).** This slave system controls access in section V_2 and part of section V_4 , with 26 doors in total, communication system and the movement sensors.
- (3) **Slave system 3 (containing CPU3 & CPU4).** This slave system has a slave subsystem, both control 193 doors in sections V_5 and V_6 ; these sections are integrated with 24 movement sensors and 193 intercoms for the communication system.
- (4) **Slave system 4 (CPU5 & CPU6).** This slave system has also a slave subsystem, both control 257 doors that correspond to sections V_7 and V_8 , respectively. They also receive the signals from 18 movement sensors and control the communication in these two sections.

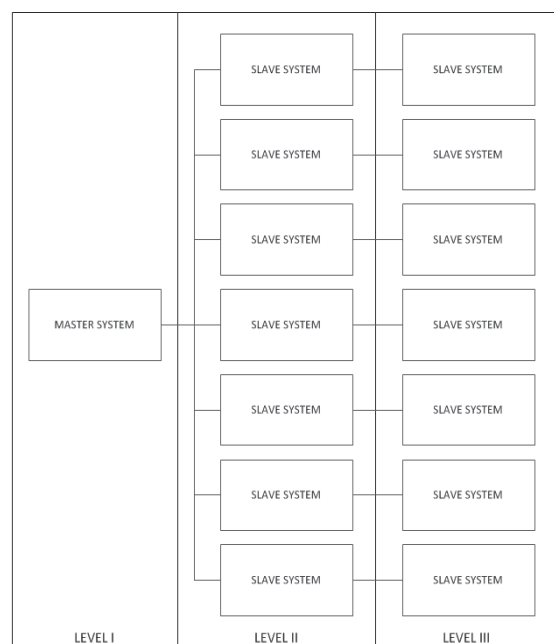


Fig. 6. Three level system

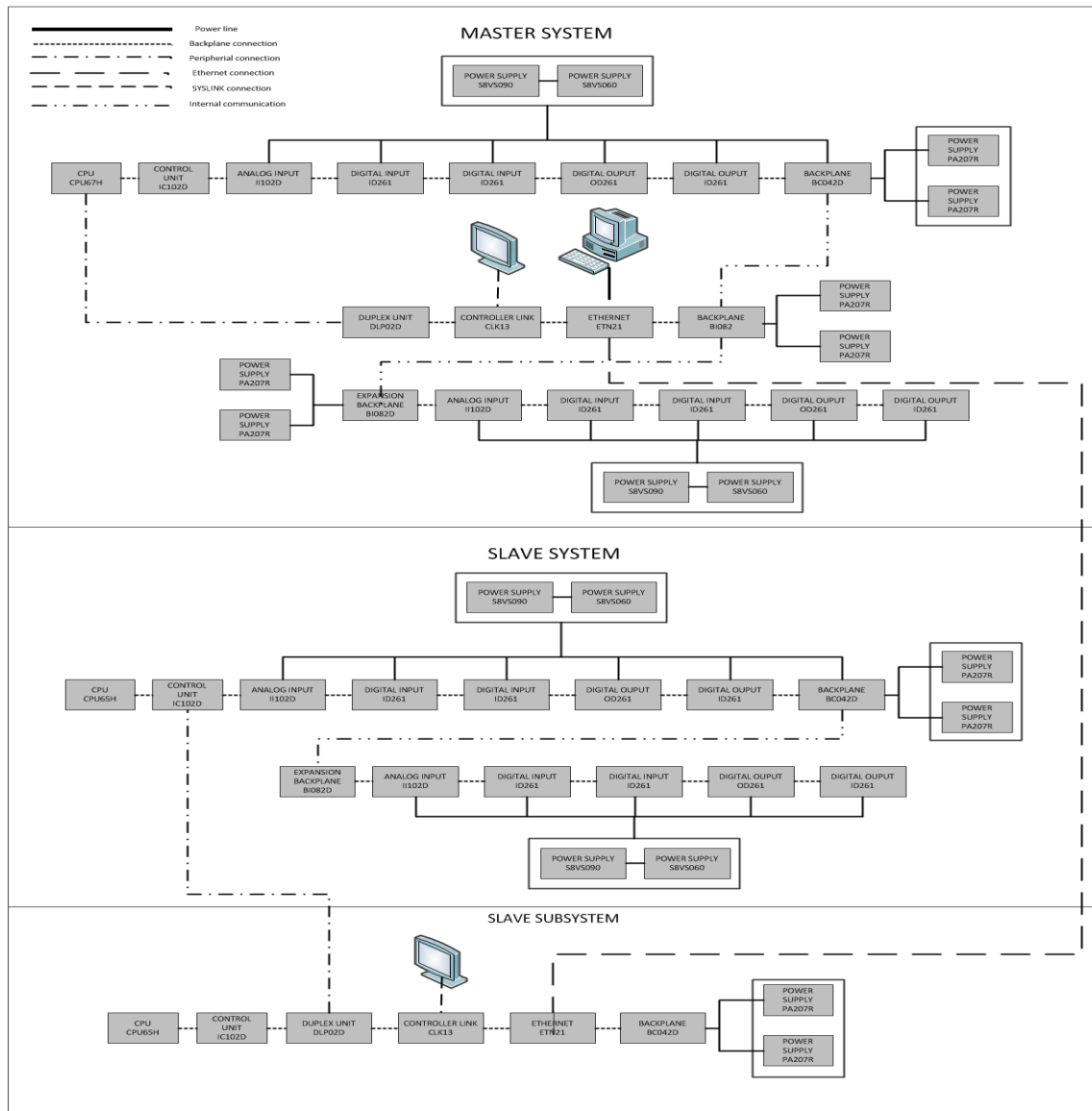


Fig. 7. Master-slave-slave system diagram

- (5) **Slave system 5 (containing CPU7).** This slave system controls 55 doors in section V_9 on the 3 levels, 12 movement sensors and 59 intercoms for the communication system, and alarm system respectively.
- (6) **Slave system 6 (containing CPU8).** This slave system controls sections V_2 and V_3 and is integrated with 31 doors. It also receives the signal from 8 movement sensors and the communications between the doors.

The automation company that proposed the design of the control system need to establish a redundant system to improve the reliability and ensure the performance of the system. However, the facility has already installed a system and is considering replacing the repaired components with the new components. In this case, the availability is used to calculate the number of components that are necessary to enhance the performance of the system.

Then, we obtained the MTTR and MTTF data based on the manufacturer's data, to calculate the reliability and maintainability, respectively. The data were provided by the manufacturer and the company that designed the system, as described

in Tables 2, 3 and 4. In this system, the only constraint is the cost of the redundancy; the engineers designed the system to have 30% of the total modules as extras to provide a backup for system performance. The total budget of the system is set to \$1,200,000.00 as a constraint. The maximum number of components that are allowed for each module is 8.

The system has three different levels of components. It is difficult to determine the optimal number of components in parallel. This calculation is to maximize the availability of the system. However, it is necessary to set the number of components at the lowest level first, then the upper levels are calculated. This system is in a master-slave configuration. However, the system is divided in 3 levels of hierarchy where the top level is a master level and the lower levels are the slave subsystems. We can re-divide this problem into three vectors that represent each level of the system. Then, we determine the lowest level of redundancy because there are no slave components in this level. Then, we determine the number of components in the second level and use the lowest level to calculate the fitness of the second level. Subsequently, in

Table 2. Safety Control System Component Function

Model	Function
CPU67H	CPU
CPU65H	CPU
BC042D	EXPANSION I/O
BI082D	EXPANSION BACKPLANE
PA207R	POWER SUPPLY
DPL02D	DUPLEX UNIT
CLK13	CONTROLLER LINK
ETN21	ETHERNET UNIT
II102D	ANALOG INPUT
ID261	DIGITAL INPUT
IC102D	ANALOG INPUT
ID291	DIGITAL INPUT
OD261	DIGITAL OUTPUT
OD291	DIGITAL OUTPUT
S8VS090	POWER SUPPLY
S8VS240	POWER SUPPLY

Table 3. Safety Control System MTTR AND MTTF

Model	MTTF	MTTR
CPU67H	260,000	4
CPU65H	271,000	4
BC042D	427,800	4
BI082D	940,000	4
PA207R	135,000	4
DPL02D	608,300	4
CLK13	206,000	4
ETN21	366,180	4
II102D	156,160	4
ID261	149,000	4
IC102D	309,640	4
ID291	941,000	4
OD261	408,000	4
OD291	111,600	4
S8VS090	135,000	4
S8VS240	135,000	4

the top level we use both results to obtain the number of components in parallel that maximize the availability of the entire system.

The data used for each component in the system are given in Tables 3 and 4.

6. Results

The program was run in MATLAB on an Intel Core^(TM) 2 Duo 3.00 Hz CPU with 4GB in RAM. Tables 5 and 6 show the best results for 50 independent runs; there were 20 variables for this problem, with a population of 100 and a maximum of 100 generations. The objective is to find the optimal number of components that are necessary for redundancy with the maximum of availability at each level of the system

Table 4. Cost and Number OF Components for Each Model in The System

Model	Price	No.
CPU67H	\$10,577.00	1
CPU65H	\$ 6,907.56	14
BC042D	\$ 2,921.16	8
BI082D	\$ 2,395.80	16
PA207R	\$ 1,028.28	48
DPL02D	\$ 3,744.84	8
CLK13	\$ 2,708.64	16
ETN21	\$ 1,515.36	8
II102D	\$ 2,921.16	16
ID261	\$ 889.68	1
IC102D	\$ 1,640.76	32
ID291	\$ 1,177.44	31
OD261	\$ 429.99	4
OD291	\$ 1,143.12	12
S8VS090	\$ 179.82	45
S8VS240	\$ 408.24	45

Table 5. Availability Results

No.	EDA	PSO	Tabu Search
1	$1 - 1.11 \times 10^{-4}$	$1 - 0.93 \times 10^{-3}$	$1 - 0.31 \times 10^{-3}$
2	$1 - 1.11 \times 10^{-4}$	$1 - 1.35 \times 10^{-3}$	$1 - 1.28 \times 10^{-3}$
3	$1 - 0.70 \times 10^{-4}$	$1 - 0.95 \times 10^{-3}$	$1 - 1.29 \times 10^{-3}$
4	$1 - 1.27 \times 10^{-4}$	$1 - 0.59 \times 10^{-3}$	$1 - 0.21 \times 10^{-3}$
5	$1 - 1.00 \times 10^{-4}$	$1 - 1.27 \times 10^{-3}$	$1 - 1.81 \times 10^{-3}$
6	$1 - 1.20 \times 10^{-4}$	$1 - 0.61 \times 10^{-3}$	$1 - 2.10 \times 10^{-3}$
7	$1 - 0.60 \times 10^{-4}$	$1 - 0.59 \times 10^{-3}$	$1 - 1.81 \times 10^{-3}$
8	$1 - 1.70 \times 10^{-4}$	$1 - 0.61 \times 10^{-3}$	$1 - 2.10 \times 10^{-3}$
9	$1 - 1.50 \times 10^{-4}$	$1 - 0.50 \times 10^{-3}$	$1 - 2.01 \times 10^{-3}$
10	$1 - 1.00 \times 10^{-4}$	$1 - 0.59 \times 10^{-3}$	$1 - 1.10 \times 10^{-3}$

and its respective cost constraint.

Recently many meta-heuristics such as PSO and Tabu search have been employed for the reliability-redundancy optimization problems. These approaches have been demonstrated to yield good solutions in the Reliability Allocation Problem (RAP). The meta-heuristics have some advantages. PSO is one of the most recent stochastic methods developed for solving optimization problems. PSO is a population-based meta-heuristic optimization technique that converge rapidly around the global optimization. TS has the ability to escape the trap of the local optimum and have good local search. TS have been employed to solve some reliability problems, such as series-parallel systems and it also demonstrated its efficiency in finding the optimal solution for the redundancy allocation. The PSO and TABU Search (TS) were used to compare and evaluate the solution by the EDA. However these algorithms never been used in real applications problems and with components that have been repaired. Tables 5 shows the availability archived by each algorithm in effective digits and Table 6 shows the total cost of the system

Table 6. Cost Results

No.	EDA	PSO	Tabu Search
1	\$1,144,067.90	\$853,186.30	\$1,177,931.45
2	\$ 982,071.94	\$855,471.19	\$1,036,990.36
3	\$1,051,356.34	\$848,393.41	\$1,200,015.23
4	\$1,112,440.90	\$894,512.84	\$944,042.41
5	\$1,098,293.40	\$862,558.47	\$848,393.41
6	\$1,124,810.52	\$821,541.16	\$777,933.41
7	\$1,054,624.02	\$822,508.81	\$848,983.87
8	\$1,017,172.80	\$826,677.75	\$730,625.92
9	\$ 984,769.67	\$849,461.44	\$790,681.59
10	\$1,029,975.31	\$838,545.44	\$1,178,889.04

Table 7. Convergence results of 50 runs

Method	Maximum (best)	Minimum (worst)	Mean	Standard deviation
EDA	$1 - 1.15 \times 10^{-9}$	$1 - 1.50 \times 10^{-4}$	$1 - 0.33 \times 10^{-6}$	4.61×10^{-5}
PSO	$1 - 1.10 \times 10^{-3}$	$1 - 1.36 \times 10^{-3}$	$1 - 0.69 \times 10^{-3}$	3.64×10^{-4}
TS	$1 - 0.29 \times 10^{-3}$	$1 - 0.43 \times 10^{-2}$	$1 - 1.14 \times 10^{-3}$	7.21×10^{-4}

achieved for each algorithm.

The PSO used for the comparison was a canonical one. The learning rate for the swarm was $c_1 = 0.738$ and $c_2 = 1.51$. The size of the swarm was set to 100 the number of informants; in other words, exploration was allowed. The TS used for the comparison the classical TS, with the Tabu tenure is set to 6 and the number of neighborhoods is set to 841. The size of the search space was set to 100.

The solutions obtained for EDA, PSO and TS for availability and cost are shown in Tables 5 and 6 for the master system and the following subsystem and its respective subsystems. The components reliability and maintainability are high; as a result, the calculation of the availability is also high. Therefore, the resulting availability is shown as a probability between 0 and 1, with significant digits. To have a better appreciation of the result, the values are presented in this form. The results in Tables 5 and 6 are the best 10 results among the 50 runs results of each method.

However, the highest system availability achieved by EDA as 0.9999 in several runs. The Tables show the cost of the allocated components in parallel for the entire system, where we observed that the highest cost is achieved by EDA. Although the cost in the PSO solution was lower than the one achieved by the EDA, this was the result of the total number of redundant components selected in the system.

Also, Tabu search achieve lower cost than the EDA due to the number of components allocated in parallel for each component in each subsystem. Increasing the number of components also increase the availability and the cost, however in EDA the components with lower maintainability have higher priority in the selection of the number of components in Parallel, and their hierarchy in the system, taking into account the importance in the system.

Statistical analysis for the 3 methods in 50 runs for the high security facility are summarized in Table 7. By using the results in Table 7, in terms of mean and best result, the solution

of EDA are better than the solutions found by PSO and TS methods for the redundancy allocation problem. In addition, the standard deviation of the results by EDA in 50 independent runs is also very small. When pursued by Behrens-Fisher's method, the t-test result supports that EDA is better than PSO and TS with significant level 1%.

7. Discussion

The EDA is an innovative method that has not been widely applied in various academic fields. It has proved effective for combinatorial problems such as scheduling problems. In this research, an EDA was used for a reliability allocation problem to maximize the availability of the system. A Gaussian univariate distribution was implemented to find the best combination of components for the high security control system. The EDA yielded good solutions due to their learning technique introduced in the algorithm by introducing a probabilistic distribution into the population. The algorithm learned from the problem structure to find the best combination for a number of components and obtained an optimal solution in a small number of generations. This paper introduced a real application with a three level system; due to the system complexity, it was necessary to modify the equation to evaluate the entire system availability by setting the lower levels' availability to find the the upper levels' availability and, as a consequence the system optimum allocation.

Compared with PSO and TS, the EDA resulted in a superior performance in terms of solution quality and CPU time. Furthermore, the random mechanisms of crossover and mutation were avoided by the EDA approach and replaced by an estimation of the probability distribution, which exploits the search of promising solutions, at each run.

Compared to the EDA, the classical PSO used in this research had low convergence speed and prematurity that made the solution to converge to local optima; deciding the values of various parameters, such as the size of the swarm, the coefficients and the number of informants, depends on the user's experience. This fact makes it difficult to tune the parameters to achieve better results.

TS is an algorithm that uses memory-based strategies. TS starts from an initial solution, and at each step, a neighbor solution is chosen to improve the objective function value. However, this algorithm does not guarantee the optimum solution, and experimental results showed that it is not successful in finding near-optimum solution due to missing promising solutions in the search space, and the lack of information exchange between neighbors.

For PSO and TS to yield good solutions, it is necessary to modify both algorithms to achieve better efficacy.

8. Conclusions

Redundancy allocation optimization problems are often encountered in complex system designs. This paper proposed a new formula that takes into account the availability, instead of the reliability, of a system.

The availability of a system with repairable components provides a measure of the performance of all the components in the system by including the failure rate from the repaired components. Therefore, the system can be designed with components that have high MTTF with respect to MTTR, and

Table 8. Notations

Nomenclature	
S	Number of subsystems in the system
n	Number of components in the subsystem
r_{ij}	Reliability of each component j in subsystem i
m_{ij}	Maintainability of each component j in subsystem i
a_{ij}	Availability of each component j in subsystem i
A	System availability
w_{ij}	Weight of each component j in subsystem i
c_{ij}	Cost of each component in j subsystem i
x_{ij}	Number of components in j subsystem i
C	Upper limit on the cost of the system
W	Upper limit on the weight of the system
$c_{MTTR_{ij}}$	Component j repair cost of subsystem i
W	Upper limit on the weight of the system
R_i	Reliability of subsystem i
M_i	Maintainability of subsystem i
a_i, b_i	Coefficient of functional relationship between manufacturing cost and MTTR of component i
pl	Probability distribution
$D_i^{Selected}$	Population of selected N individuals in generation l
$evalf(x)$	Fitness function for EDA

maintenance time can be reduced by selecting elements with low MTTR.

In real life applications, systems are not only composed of new components but also repaired ones. In this case, the reliability is not a suitable measure for repaired components, and it is necessary to take into account the failure rate. A new series-parallel redundancy allocation mathematical model is formulated to maximize the system availability. This mixed integer problem is difficult to solve even by the heuristics method; therefore we proposed the EDA approach as a recently developed meta-heuristic method. We applied the EDA, PSO and Tabu Search algorithms, separately, to solve a high-security system redundancy allocation optimization problem. The optimal solution for this problem demonstrated that the EDA yields an optimal system availability that is better than the PSO and TS results, because the EDA learned from the structure of the problem. For future work, a more complex applications will be considered. Although the RAP can also be considered as a multi-objective problem, the EDA could be used to find the optimal set of solutions for this problem.

References

- (1) A.D. Kiureghian, D.O. Dalager, and J. Song: "Availability, reliability and downtime of systems with repairable components", *Reliability Engineering and System Safety*, Vol.92, No.2, pp.231–242 (2007)
- (2) B.R.B. Abu and J. Watada: "Biological computation of optimal task arrangement for a flexible machining cell", *Journal of Systems and Control Engineering*, Vol.222, No.6, pp.605–618 (2008)
- (3) B. Shumeeet and C. Rich: "Removing the Genetics from the Standard Genetic Algorithm", Morgan Kaufmann Publishers, pp.38–46 (1995)
- (4) B. Richard and P. Frank: "Theory of maintained systems: distribution of time to first system failure. Math. operat. Res.", Prentice-Hall, pp.32–42 (1977)
- (5) B. Benjamin, V. Dinesh, and P. Elmer: "Maintainability: A Key to effective Serviceability and Maintenance Management", Prentice-Hall, pp.1–26 (1997)
- (6) C. Tianshin, T. Ke, C. Guoliang, and Y. Xin: "On the analysis of average time complexity of estimation of distribution algorithms", in Proc. IEEE Congr. EVol. Comput. (CEC), pp.453–460 (2007)

- (7) C. Tianshi, H. Jun, S. Guangzhong, C. Guoliang, and Y. Xin: "A new approach to analyzing average time complexity of population-based evolutionary algorithms on unimodal problems", *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, Vol.39, No.5, pp.1092–1106 (2009)
- (8) C. Benhui and H. Jinglu: "A hybrid EDA for protein folding based on HP model", *IEEJ Trans. Electrical and Electronic Engineering* Vol.5, No.4, pp.459–466 (2010)
- (9) C. David and K. Abdullah: "Multiple Weighted Objectives Heuristic for the Redundancy Allocation Problem", *IEEE Trans. Reliability*, Vol.55, No.3, pp.551–558 (2006)
- (10) C. David and A. Smith: "Reliability optimization of series-parallel systems using genetics algorithm", *IEEE Trans. Reliability*, Vol.45, No.2, pp.254–260 (1996)
- (11) F. David, H. William, and L.N. Kee: "System reliability allocation and a computational algorithm", *IEEE Trans. on Reliability*, pp.64–69 (1968)
- (12) K.K. Govil: "Maintainability and Availability calculations for series, parallel, and r-out-of n configuration", *Micro-electron, Reliability*, Vol.23, No.5, pp.785–787 (1983)
- (13) N. Jack: "Analysis of a repairable 2-unit parallel redundant system with dependent failures", *IEEE Trans. Reliability*, Vol.R-35, No.4, pp.444–446 (1986)
- (14) J. Ying-Shen, L. Shui-Shui, and H.-P. Kao: "A knowledge management system for series parallel availability optimization and design", *Expert Systems with Applications*, Vol.34, No.1, pp.181–193 (2008)
- (15) K. Ino and J. Watada: "Decision making with an interpretive structural modeling method using a DNA-based algorithm", *IEEE Trans. on Nanobioscience*, IEEE Publisher, Vol.8, No.2, pp.181–191 (2009)
- (16) P. Larranaga and J. Lozano: Estimation of Distribution Algorithms: New Tool for Evolutionary Computation, Boston: Kluwer Academic Publishers (2002)
- (17) L. Xianneng, M. Shingo, and K. Hirasawa: "An extended probabilistic model building genetic network programming using both of good and bad individuals", *IEEJ Trans. Electrical and Electronic Engineering* (2013) DOI: 10.1002/tee.21864.
- (18) M. Eghbal, N. Yorino, Y. Zoka, and E.E. El-Araby: "Application of Multi-Objective Evolutionary Optimization Algorithms to Reactive Power Planning Problem", *IEEJ Trans. Electrical and Electronic Engineering* (2009) 4: 625632
- (19) R. Prasad and M. Raghavachari: "Optimal Allocation of Interchangeable Components in a Series-Parallel System", *IEEE Trans. Reliability*, Vol.47, No.3, pp.255–260 (1998)
- (20) S. Shripprakash, K. Dinesh, and K. Ajay: "Availability optimization of a series-parallel systems using genetic algorithms", in Proc. XXX National systems conference, NSC 208, pp.640–644 (2008)
- (21) Stapelberg Rudolph Frederick, Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design", Springer (2009)
- (22) S.C. Sup and C.Y. Kwon: "Branch-and-Bound Redundancy Optimization for a Series System with Multiple-choice constraints", *IEEE Trans. Reliability*, Vol.48, No.2, pp.108–117 (1999)
- (23) W. Bo, L. You, and J. Watada: "Re-Scheduling of Unit Commitment Based on Customers' Fuzzy Requirements for Power Reliability", *IEICE Transactions D*, Vol.94-D, No.7, pp.1378–1385 (2011)
- (24) Y.S. Bahar, J. Watada, and J. Fulcher: Structural learning of the Boltzmann machine and its application to life cycle management, *Neurocomputing*, Vol.74, No.12-13, pp.2193–2200 (2011)
- (25) Y. Wei-Tzen and C.C. Lin: "A reliability model for dependent failures in parallel redundant systems", *IEEE Trans. Reliability*, Vol.R-23, No.3, pp.286–287 (1974)
- (26) Z. Leila and C. David: "Redundancy Allocation for Series-Parallel systems using a Colum Generation approach", *IEEE Trans. Reliability*, Vol.59, No.4, pp.706–717 (2010)

Appendix

Haydee Melo was supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) scholarship program for international Students. This work was supported in part by Grants-in-Aid for Scientific Research, MEXT (No.23500289).

Haydee Melo (Non-member) received the B.Sc. degree in Electromechanics engineering from National Polytechnic Institute (IPN), Mexico and the M.Sc. degree from the Graduate School of Information, Production and Systems, Waseda University, Japan. She is currently a Doctor student in Graduate School of Information, Production and Systems, Waseda University, Japan as a scholarship recipient student from the Ministry of Education, Culture, Sports, Science and Technology, Japan. Her research interests include soft computing, management engineering, reliability and Meta-heuristic algorithms.



Junzo Watada (Member) received his B.S. and M.S. degrees in electrical engineering from Osaka City University Japan, and Dr. of Engineering on “Fuzzy Multi-variant Analysis and Its Applications” from Osaka Prefecture University, Japan. His professional interests include soft computing, tracking system, knowledge engineering, and management engineering. Prof. Watada is a recipient of the Henri Coanda Gold Medal from Invention in Romania in 2002. He is a Life Fellow of Japan Society for Fuzzy Theory and intelligent Informatics (SOFT) and Biomedical Fuzzy System Association (BMFSA). He is the Principal Editor, Co-Editor, or Associate Editor of various international journals, including the International Journal of Biomedical Soft Computing and Human Sciences, ICIC Express Letters, the International Journal of Systems and Control Engineering, and Fuzzy Optimization and Decision Making.

