INTEGRATED PERFORMANCE-RELIABILITY OPTIMISATION OF SYSTEMS WITH MULTI-LEVEL REDUNDANCIES

OKACHI IKEGWURU IBEZIMAKOR

A thesis submitted in partial fulfilment

of the requirements of the
University of Northumbria at Newcastle
for the degree of
Doctor of Philosophy

Research undertaken in the
Faculty of Engineering and Environment

October 2016

Abstract

Redundancy allocation, in the context of reliability driven design, is the process of multi-objective optimisation of system configuration with reliability and cost related objectives. Large systems, of any type and discipline, can be divided into several subsystems comprising modules and components. Such a hierarchical form of system arrangement is regarded as multilevel configuration. These systems have the performance capability beyond traditional binary reliability framework of either completely working or totally failed. Large systems normally have redundancies at different levels. In current practice, multi-level redundancy allocation takes place sequentially. This is mainly due to lack of a robust optimisation method capable of delivering large scale redundancy allocation problems. Development of such methods leads to design of enhanced systems with better performance in terms of cost and reliability. The overall aim of this project is to develop a method for multi-state reliability optimisation of large real-world systems. To achieve the overall goal, firstly, a genetic algorithm (GA) suitable for analysis of systems with multi-level redundancies is developed. For this GA, new multi-level chromosome, new crossover and mutation operators capable of combining building blocks at different level and mutation of solutions at various levels are designed. Whilst the GA chromosome and regeneration operators are specially designed for handling multi-level systems, in the second step a Non-dominated sorted genetic algorithm (NSGA-II) is developed for multi-dimensional search towards finding Pareto frontier solutions with respect to a number of cost-related, performance-related and reliability-rated objectives including cost, size, weight, availability and failure rate. In the final stage, the developed search and optimisation methods are implemented in a software tool written in MATLAB. Employing the optimisation tool for benchmark problems with multi-level redundancies, heating, ventilation and air conditioning (HVAC) systems, it has been shown how an integrated multi-level redundancy allocation, as opposed to sequential redundancy allocation, can lead to superior solutions.

Table of Contents

List of Figures

Acknowledgement

My sincere gratitude goes to **God Almighty** for the gift of life and his extra ordinary grace towards me to successfully finish strong, especially after series of unfortunate circumstances that greeted my family in the final stage of this research. Thank you God, and please let me not be best identified by this degree, instead by your priceless grace. Your grace is supreme!

I would like to specially thank my Supervisor **Dr. Alireza Maheri** for his enormous contributions and commitments towards the success of this work. It is worth thanking him for accepting to supervise me in the first place having known my technical deficiency from the outset, yet he signed in for it, and today he has left me better than he met me. **Dr. Alireza Maheri** remained a source of inspiration to me whilst this doctorate degree last both academically and socially; his supervision developed me into been a more competitive individual with good thinking and problem solving skills.

I would also like to graciously thank **Dr. Martin Birkett** for proof-reading my thesis under his tight schedule and informing me with his valuable comments for improvements.
Let me gladly thank my family, particularly my dear mother Mrs. **Eunice Wori Okachi** for her motherly advice, moral discipline and encouragement to me all through my life.

My sincere gratitude goes to my friends, my benefactors and other financial organisations that did assist me in my most difficult moments, especially **Engr. & Mrs. Echeonwu Isidore Azubuike,** my best friend and my dearest life companion **Miss. Sophie Amaka Nmabulo Jacinta Opene.** I thank you all; and just to let you know that the exceptional motivation and encouragement you individually and collectively supplied to me throughout my doctoral training was full of zest, and were simply phenomenal.

To my upright uncle, my proven benefactor and his dear wife **Mr & Mrs Okachi Everest Obasinachi,** you paid the highest price to bring this dream to reality. You reposed your confidence and trusted me even when l had less trust to myself as it were, to encourage me to go in for a PhD degree. As eve that was not enough; you progressively humbled by doubts through your fervent promise which you kept, to shoulder the financial implications leading to the success of this research under any circumstance. To me, this is what matters most in my career. Thank you Uncle.

Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others.

Name: OKACHI IKEGWURU IBEZIMAKOR

Signature:

Date: 11 October 2016

Notations

A	Module.
<n, m>	Min/Max number of module components.
A1 <n, m>	Module A1 with min/max components.
	Indicates the presence of system branch or sub-branch.
U_S	System level
R_i	Reliability of U_i
U_i	i-th unit; a common name for system, module and component.
$U_{i,m}^{j}$	j –th redundant unit of m-th sub-unit of U_i
$R_{i,m}^{j}$	Reliability of $U_{i,m}^{j}$
x_i	Number of components used in unit U_i
$R_i(\mathrm{x}_i)$	Reliability of subsystem i
n_i	Number of sub-units in U_i

Acronym

ASS	Availability of a system steady state
DFM	Double failure matrix
DM	Decision maker
FR	Failure rate
GA	Genetic algorithm
HGA	Hierarchical genetic algorithm
Iff	If and only if
MDT	Mean down time
MFOP	Maintenance free operating period
MLF	Multilevel formulation
MLMOR2	Multilevel multi-objective redundancy optimisation
MTBF	Mean time between failure
MTBM	Mean time between maintenance
MTTF	Mean time to failure
ngen	number of generation
npop	Number of population
NSGA-II	Non-dominated sorted genetic algorithm
pc	probability of crossover
PFD	Probability of success or probability of failure on demand
pm	probability of mutation
RAP	Redundancy allocation problem
RBD	Reliability block diagram
VEGA	Vector evaluated genetic algorithm

1. Introduction

Properly designed systems are mostly justified through their operational capabilities. Such systems operational capabilities have the tendency to successfully execute tasks in time, and under optimal performance without failing. No system is practically designed not to fail. However, failure of any system without accomplishing its assigned task defines such system not only nonperforming and unreliable, but also a risk to safety assurance. Safety prevents systems from causing injury to persons, or significant material damage during its use. A good system is a one designed with safety assurance. Therefore, every safety assured system is usually subjected to safety evaluation. That is, to evaluate safety that encompasses the following aspects: safety when the system is functional and is correctly operated and safety when the system or part of it has failed. The former is concerned with accident prevention, for which a large number of regulations exists both nationally and internationally. The later aspect deals with that of technical safety which is usually checkmated in five steps (identification of potential hazards, identification of their causes, determination of their effect, classification of their effect, and investigation of all possibilities to avoid hazard). Considering similar tools with respect to reliability is however different, and literally very necessary to establish the distinction between safety and reliability. Safety deals with measures that allow a system to be brought into a safe state in the case of failure (fail-safe behaviour), while reliability is concerned with performance measures aimed at minimizing failures. This is based on the fundamental belief that a system is said to be reliable and safe only if it is able to successfully execute a given task within a given time frame without failures. Failures can lead to loss of lives, lost production, or can even be costly due to damage to components in several industries including but not limited to the oil and gas, and aerospace industries (Branimir *et al.,* 2016). For oil and gas industry, particularly production systems, major component failures amount to lost time,

and can also lead to loss of lives if harmful chemicals are released into the atmosphere. Such a scenario is similar in the aerospace industry where the cost of failure is also predominant, and in most cases it becomes highly impossible to recover. Therefore, the practice of reliability on systems remains an issue of concern in the engineering industries which are targeted at addressing failure criticality alongside with the costs associated.

To design a reliable system capable of performing highly and at the same time guarantee safety is therefore important. In a nutshell, reliability, performance and safety are very fundamental objectives to consider in the engineering design and they deserve to be well understood. Having stated the safety concept and defined reliability above, performance on the other hand is characterised by the amount of useful work accomplished by the system. In most cases however, system performance can settle at binary state; that is either performing or failed depending on the type of system and the nature of design implemented on it. For other systems such as power generation, HVAC, flight control system and oil and gas transportation, the overall performance can settle on different levels (e.g. 100%, 80%, and 50% nominal capacity). Therefore, to have a reliability driven system that can perform optimally and safely the design objectives must begin at the early design stage where all the necessary factors such as design approach and design parameters are suitably dealt with. Objectives that drive design could be conflicting in most cases and the ability to trade-off the conflicts results in good system design. For instance, a design could be driven by reliability as the main objective with other assessment criteria, including the cost, as the secondary objectives given the potential cases; (i) when a failure leads to huge financial loss or human lives and (ii) in case of dealing with a great deal of uncertainties in modelling and evaluation.

1.1 Basic Quantitative Reliability Requirements

Quantitative reliability requirements in engineering systems for example HVAC and Flight Control systems are mainly based on reaching high availability targets. Given this, a small risk of failure associated with occurrence of premature failure is hardly guaranteed. At a very high availability, the probability of a premature failure can lead to an increase in cost of the failure or total loss of the system (warranty cost for example). Overall, reliability requirements must endeavour to guarantee not only a high availability target, but also a low probability of premature failure.

Considering cost, it is normally a factor often considered as one assessment criteria in terms of price for the achievement of a desired reliability level of a given system. This can represent an allocated capital cost (budget) which involves the cost of implementing and operating a program, which is normally in addition to the overall development and production cost associated with the system. It literally consists of direct material and labour costs as well as indirect costs such as, taxes, insurance, energy, production facilities and equipment. Other costs are overheads such as administrative, marketing and product development costs. With regards to capital cost, it is generally (but not always) an increasing function of reliability. However, capital cost can be viewed as an investment for achievement of a desired system reliability. This is largely so because organisations that put more resources into a system expect a better reliability result. Cost and reliability can therefore be expressed mathematically to show their relationship.

Mathematically, overall system reliability R_s , is related to component reliabilities R_i , through a function given by;

$$R_s = R_1 \times R_2 \times R_3 \times \ldots \times R_m \tag{1.1}$$

where $R_1, R_2, R_3, \ldots, R_m$ are reliability of the system components.

(Where $1 \leq i \leq M$ number of components in a system)

Assume $C_i R_i$, to be the cost of component i, with reliability R_i, implying that the cost increases with increasing component reliability. So the total system cost C_{total} can be demonstrated as;

$$C_{total} = \sum_{i=1}^{M} C_i(R_i) \tag{1.2}$$

Considering the relationships in Equations 1.1 and 1.2, the common objectives of already established reliability optimisation approaches term to be:

- For a pre-defined level of system reliability, minimize the total cost of resources required to attain a desired reliability level.
- For a given budget allocated for a particular project, achieve maximum level of system reliability.

As regards to the first case, the objective is to determine the optimal reliability allocation such that the reliability of the system R_{Min}, and the total system cost C_{total} are minimised. Based on this, the following optimisation problem becomes:

$$\min_{\{R\}_1^M} J = \sum_{i=1}^{M} C_i(R_i) \tag{1.3}$$

subject to the constraints

$$(R_1 \times R_2 \times R_3, \ldots, R_M) \geq R_{Min} \tag{1.4}$$

where,

$$\{R_s\}_1^M = \{R_1, R_2, R_3, \ldots, R_M\} \tag{1.5}$$

As regards to the second case, the objective requires the designer to determine the optimal reliability allocation that maximises the system reliability R_s, subject to the overall cost C not exceeding some Pre-specified budget value C_{Max}. Then, the optimisation problem appears in the following:

$$\max_{\{R\}_1^M} J = (R_1 \times R_2 \times R_3,..., R_M) \tag{1.6}$$

Subject to the constraints

$$\sum_{i=1}^{M} C_i(R_i) \leq C_{Max} \tag{1.7}$$

$$with \qquad 0 \leq R_i \leq 1 .$$

1.2 Generalised Approach of Reliability Goals Establishment

Konya (2012) stated that establishing reliability goals is a crucial aspect of achieving and maintaining acceptable reliability performance. They are developed to:

- Establish product-level reliability specifications that if met, shall directly ensure that the reliability performance of the product will meet the customer's functional needs and be consistent with other product constraints.
- Allocate the product –level reliability requirements down to the level needed (for example, subsystems, equipment or assembly level) in order to be very meaningful to the design engineers.

These goals are however in contrast with reliability requirements. Whilst reliability requirements remain the minimum level of performance expected by the customer, reliability goals deal with higher level reliability that surpasses reliability requirement of a system. This

may result in one or more of the following benefits: a greater market share (i.e., a situation where reliability takes a centre stage and obviously becomes a discriminator among products), lower life cycle cost for the customer, lower supplier cost (fewer returns and lower warranty costs), and less risk of reliability and litigation (safety cases). Ideally, reliability requirements and goals are normally defined clearly prior to the beginning of the design or product development circle. This is to address the resemblance of these two objectives (requirements and goals). At the initial design stage, requirements may be stated as "goals" depending on when firm requirements can be defined as the design matures. Additionally, reliability goals and requirements can be based on customers' preferences. Alternatively, it can be on internally developed strategies to position new systems in the open market environment (i.e., competitive advantage through reliability benchmarking). Reliability requirements and goals are, over time, kept in check in order to realise them within schedule constraints and budget. The following are fundamental to realise reliability goals and requirements:

- Modelling and Simulation
- Benchmarking
- Fault tolerance
- Environmental Characterization and
- Durability Assessment.
- Model-based and observer-based fault detection.

Model-based and observer-based fault detection method uses the dynamical model of the system to generate signals (called residual signal) to give an indication whether a fault has occurred in the system (Ding *et al.*, 2016). Here in this work, the candidate shall not be concerned with the aforementioned method. Rather, the candidate shall concentrate on using MATLAB program for successful delivery of this work.

1.3 Research Aim and Objectives

The overall aim of this research is to establish a method for multi-state reliability optimisation of large real-world systems where an integrated method for multi-level redundancy allocation of such systems is realised. These systems normally have redundancies at different levels. Upon realisation of this aim, the aforementioned systems performance are maximised, their failure rates, weights and costs, depending on the objectives, are minimised (Zhigang & Ming Zuo, 2006). Recent research carried out in this area mainly proposed a multi-objective optimisation model for sequential redundancy allocation. This model can be time consuming and solutions achieved through this model are usually not optimal.

1.3.1 Research Objectives

Objective 1- Development of a genetic algorithm (*GA*) suitable for analysis of systems with multi-level redundancies

This is to technically avoid using the conventional method of chromosome definition for systems with multi-level redundancies, because it leads to very long chromosomes for real-life systems. Such long chromosomes usually have a diverse effect on the performance of a *GA* in optimal redundancy allocation.

Therefore, a suitable *GA* for multi-level reliability systems should have the following characteristics:

i. Multi-level chromosome representing systems with multi-redundancies.
ii. A new crossover operator capable of combining building blocks at different levels.
iii. A new mutation operator capable of muting a solution at various levels.

Objective 2 – Development of multi-objective codes suitable for multi-level systems and tailoring of the *GA* operators developed in objective 1 into a higher *GA* level called a non-dominated sorted genetic algorithm (*NSGA-II*) for size optimisation.

Objective 3 – Formulation of typical systems with multilevel redundancies, applicable to large real-world systems such as flight control systems (*FCS*), hybrid renewable energy systems (*HRES*), and heating ventilation and air-conditioning systems (*HVAC*).

Objective 4 - Run case studies towards evaluating the performance of different systems designed using traditional binary-state methods and new multilevel formulation method.

1.4 Original Contribution to Knowledge

This project contribution to knowledge is the development of a suitable method for multi-state reliability optimisation as opposed to the traditional binary performance case. Development of such a method leads to design of enhanced systems with better performances with respect to both reliability and cost. Integrated multi-level redundancy allocation is an additional contribution of this project to knowledge, hence it is less time consuming as opposed to the sequential method mentioned earlier. Moreover, the *GA* operators are innovative and can combine building blocks at different levels.

1.5 Structure of the Thesis

The rest of the thesis is structured in the following ways:

Chapter two deeply emphasized on maintenance and maintainability importance of typical real-life systems. Various parameters used for measuring reliability of such systems were analysed. The chapter extensively discussed on reliability concepts to show that reliability drive design is a function of time. In general, reliability of systems is more effective at their early stage of life but decreases with time due to component aging, failure and the use of substandard

materials during design stages. Since addressing unreliability is one of the concerns in the engineering field, the chapter established that a good design approach capable of addressing physics of failures at every design stage should be adopted. Furthermore, the chapter elaborated on various ways systems can fail and what leads to system failure. Often the cause of failure originates from system faults and progressively results in system failure if not promptly identified and fixed. Finally, the chapter showcased up to date types of system configurations and modern analysis techniques by which systems can be effectively analysed, alongside ways in which system design for reliability can be actualised.

Chapter three of the thesis is structured to critically explore redundancy allocation problems (*RAPs). RAPs* are generally a multi-objective optimisation problem *(MOOP*), so the chapter addressed RAPs with the *MOOP* approach. Apart from that, the chapter presented a variety of approaches and techniques of *MOOP* and established that the *NSGA-II* method was most suitable to adopt for successful delivery of this project as the approach is flexible, gives a better spread of solutions, gives a good level of convergence and can allow room for solution improvement.

In Chapter four, solutions to *RAPs* were provided and analysed against the traditional approach where redundancies were allocated on systems sequentially. The new approach is efficient and is referred to as an integrated approach and enables redundancies to be allocated on systems automatically without limitations on either of the system levels (i.e., components, module, branch, sub-branch and system).

Chapter five presents the usability of the *MATLAB* software development tool used in this work with its functionalities explained. The chapter shows that *MATLAB* is a prolific software tool for technical project delivery. From projects conceptual stage, particularly projects involving Series-Parallel systems, the tool enables the designer to configure design candidates in

accordance with specification to produce the desired output. The program worked efficiently well with a very short run time required to produce desired outputs or solutions. In this case, desired outputs were shown in two phases:

- Showing system topology levels with integrated allocated redundant components which outweighs the traditional design approach on the redundancy allocation problem
- Showing the multi-objective results on the Pareto frontier for decision making.

Chapter Six showcased various case studies proving successful application of the written *NSGA-II* codes which includes the following:

- Optimisation of fuel oil system design with creation of more redundancies up to multilevel using *NSGA-II* system configuration mechanism.
- Increasing Series systems overall topology level by incorporating redundancies at all levels with coded solution version provided and analysed.

Chapter seven is conclusions and recommendations.

2. Reliability Driven Design Concepts and Analysis Methods

2.1 Basic Reliability Concepts

Reliability concept is very important for many industries. The concept further took pervasive dimension, as a fundamental measure and practice worthy of both qualitative and quantitative connotations (Marseguerra & Zio, 2009). Interestingly, reliability is mainly concerned with dependability in the life cycle management of systems. In other words, the probability that a system will perform properly under a given operating time, *t*, (Lewis E.E., 1996). Reliability of a system is a function of time, $R(t)$, and as earlier stated, it is concerned with dependability and excellent performance that enables the principle necessitate: a dynamic and probabilistic framework. It is measurable in terms of a probability, $P(T > t)$ that a system or a component will continue to function optimally under a defined time interval, t prior to failing eventually in time T, such that T, is regarded as a continuous random variable. $T > t$. For this purpose, it is known as survival of function (Kaufmann *et al.*, 1977).

$$R(t) = P(T > t) \tag{2.1}$$

The relationship above is restricted to the time interval of the system initial failure for non-repairable systems, whilst all total time intervals between successive failures must be considered for repairable systems. Overall, the reliability function is treated as a monotone decreasing function with an appreciable unity life at the start, which reduces gradually towards zero as the time increases to infinity.

The component of $R(t)$, from the above Equation (2.1) is the cumulative distribution function of failures $F(t)$, and is linked with reliability in accordance with the relationship below.

$$F(t) = 1 - R(t) \tag{2.2}$$

Similar to the adopted approach in Equation(2.2) , and continuous random variable T , the $F(t)$ gives the probability that the time to failure T will be smaller than the specific time t ,

$$F(t) = P(T \leq t) \tag{2.3}$$

The probability density function of failure time, or the time to failure is denoted by $f(t)$, and it normally describes the spreading nature of failure probability. In the infinitesimal interval t , $t + dt$, probability of failure is defined as $f(t)dt$, while for any specified interval of time $t_1 \leq T \leq t_2$, failure probability can be estimated as:

$$P(t_1 \leq T \leq t_2) = \int_{t_1}^{t_2} f(t)dt \tag{2.4}$$

The reason is because $f(t)$, is a probability distribution whose values are always non-negative, and the total area beneath $f(t)$, is always equal to one. Also, $f(t)$, has a relationship with the cumulative distribution function $F(t)$, as shown below;

$$f(t) = \frac{dF(t)}{dt} \tag{2.5}$$

Further fundamental discussions relating to basic reliability concepts, technicality and the applicable mathematics can be found in (Technical Manual 5-698-1, 2007 & Vayrynen *et al*., 2011).

2.2 Safety Systems

The ideal concept of safety systems is to develop a system free from danger to both environment and human life (Story, 1996). This concept has two fundamental requirements: a functional and non-functional requirement that assesses its dependability and design. The

functional requirement deals with activities that allow the system to execute its intended function, while the non-functional requirement deals with the basic properties such as size, weight, maintainability and many more characteristics a system should possess for adequate performance. Non-functional Requirements (NFRs) may have a profound impact on system architecture, hardware and software development process and may even impact functional safety (Petrov *et al.*, 2013). Almost all industries have safety challenges; they regard such challenges as serious and critical because the challenge increases as technology evolves. One such example is an aircraft or nuclear power plant that overtime evolves and causes a hazard as a result of a safety gap (Acharyulu, 2015). These aforementioned systems can fail and their failure is often accompanied with consequences leading to loss of life, property destruction, financial loss and environmental harm. However safety dependency on systems can be said to be less absolute, hence no best design approach has been found to address safety gaps completely. Instead, safe systems only minimise a hazard, failure and the acts leading to failure which includes human errors (Naderpour *et al.*, 2015). Since the key to system safety has to do with the ability to successfully manage hazards, achieving success therefore involves systematic approach to the management of risk where all hazard can be tracked continually until acceptable closure action is verified and implemented as shown in Figure 2-1.

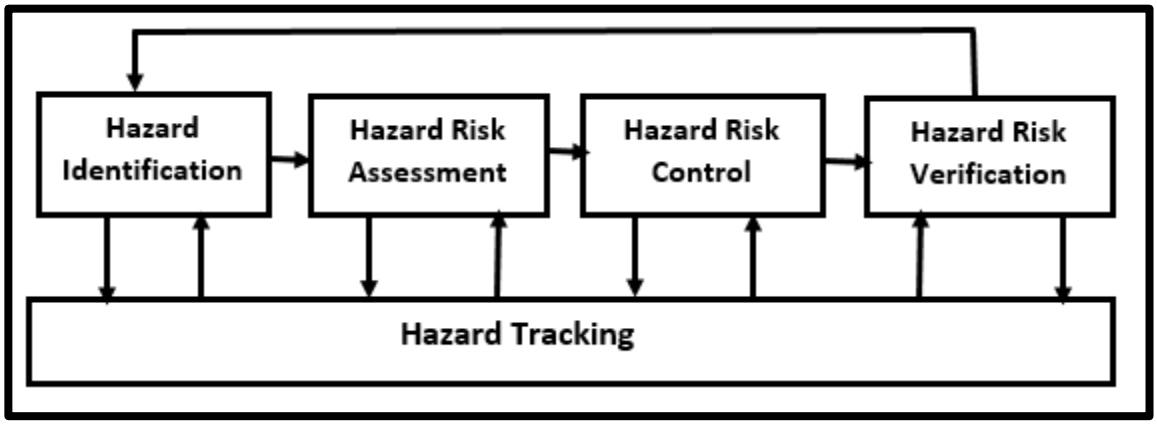


Figure 2-1 Systematic Approach to System Safety (Ericson, 2005)

2.3 System Hazard

Hazard on systems can be conceptualised as a source of energy that can be potentially released to cause injury especially in a work environment. Storey (1996) referred to a hazard as the occurrence of an unintended event or sequence of events that causes death, injury, and environment or material damage. However, there is one fundamental hazard-related concept that seems conflicting in the management of hazards known as mishap. Hazards and mishaps are at the opposite ends of the same entity. As stated earlier, a hazard is a potential event while a mishap is actual an event informed by a hazard. A hazard is predictable and can be managed effectively (Cheng-Wu *et al.,* 2012). There are different types of hazards such as natural, technological and societal hazards (Martina *et al.,* 2015) and an adequate model and estimation can be found (Nima *et al.,* 2009).

2.4 System Fault

Fault and failure have one feature in common which is that they both can obstruct safety of system. In addition, can affect efficiency and performance of a system. A fault can be said to be deviational and in many possible cases, does not keep a system out of operation. For failure, it is a permanent deviation that keeps a system out of operation until repairs are carried out. A fault has the capacity to initiate a system failure if not detected early and fixed. A Fault could be hidden and difficult to detect because it occurs independently on a system without regards to the system state, whether the system is in operation or not. Often it occurs in various forms: design faults, manufacturing faults, assembling faults, normal operation faults, maintenance faults, hardware faults, software faults etc. However, system faults can be diagnosed in order not to initiate system failure through a procedure that includes two fundamental steps: fault detection and fault localization (Maitreya & Adershpal, 2007). Lu Lu *et al*., (2013) explained that fault detection particularly identifies fault occurrence on systems. In this step, detection tools are adopted to detect the presence of system faults and whether faults truly exist. Where

faults exist, then fault localization is subsequently triggered to identify the exact fault domain and reason why the fault existed. Since fault detection is the first key step in performance of fault diagnosis on a system, ensuring accurate fault detection methods such as the classical fault detection method and passive fault detection method, amongst others, is therefore fundamental.

2.5 System Failure

While failure remains a permanent deviation that keeps systems out of operation until repairs are carried out, it is also one of the acceptable measures for reliability. The rate at which a system fails determines if such a system can be reliably enough to finish an assigned task within a time frame. In some cases, a system that fails recursively is either aging or designed with sub-standard materials. Two types of failures are well established in the literature: Common cause and Independent failures. Common cause failures are failures of multiple components due to a shared root cause (Wang *et al,* 2014; Rejc & Marko, 2014; and Ramirez-Marquez & David, 2007), while independent failures are failures of a single component which does not affect other system components from continuing in its functional state. Furthermore, the probability of failure helps to estimate accurate engineering design points (Cho, 2013). For this purpose, two reliability methods: First Order-Reliability Method (FORM) and Second Order Reliability Method (SORM) are applied. FORM attempts to approximate failure probability through linearization of the limit- state surface (the boundary of the failure domain) at design point, while SORM uses quadratic surface fitted at the design point to effect the improvement (Kiureghian & Dakessian, 1998), and tries to identify failures.

2.5.1 Causes of Failure on a System

The cause of components failure especially in the case of flight control systems, mechanical and most electronic systems can vary. In most cases the components of these systems fail when

(i) they are exposed to a high temperature, (ii) loads applied exceeds its strength, or as a result of aging and many more. Alternatively, failure can be characterised by the following factors:

- Problems in part design or manufacturing,
- Problems in the system design or assembly
- Incorrectly performed repairs, and
- Wear out resulting from cumulative usage of the system (Tsai-Ching & Wojtek, 2010).

The first category occurs early in the life of a component and their frequency, called failure rate (FR) or hazard rate, tends to decrease over time. Failure analysis can be done through modelling and identification of failure portions of a system under operating condition as depicted in Figure 2-2.

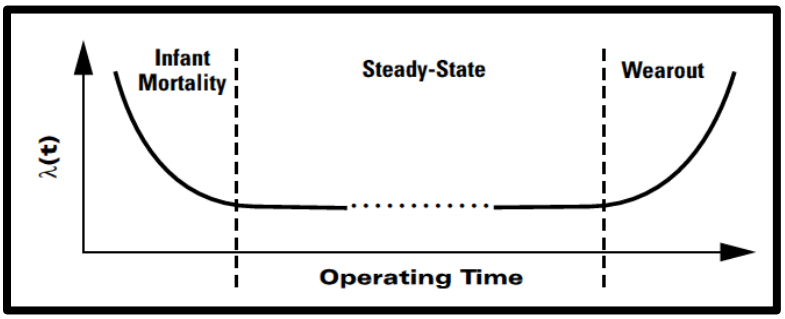


Figure 2-2- Bathtub curve FR model (Crowe *et al.*, 2000).

The curve is segmented into three periods: infant mortality, steady-state and wear out. The infant mortality period objectifies a small portion of population with initial high failure probability as a result of possible manufacturing defects that failed to show-up immediately during screening. The steady-state period represents a failed population with a constant FR, and the wear out period represents the end of life, and occurs when the failure increases in time. Failure is difficult to completely designed out from systems. For this reason, every system is therefore prone to failure. Failure occurrence literally has unavoidable effects on systems. To address failure effect, particularly the common cause failure with the potential of triggering,

failure at all levels has been handled (Chen *et al*., 2015). Failures caused by common internal cause are referred to as propagated failures and can be easily classified as local failures on the component. In general, system can fail safely or dangerously. Failure that occurs safely is otherwise known as fail-safe, while failure that occurs dangerously is referred to as fail-to danger as detailed below.

2.5.2 Failure to Danger versus Failure to Safety

Failure to danger is catastrophic, the consequences are always capable of subjecting systems to extreme destruction. Often cases of failure to danger lead to loss of lives and valuable properties. On the other hand, failure to safety neither results in loss of lives nor in loss of valuable properties. Failure to safety is basically failure detected ahead of time, thereby allowing time for precautionary measures to be taken in order to modify the design for the purposes of reducing the failure probability and its effect from failing to danger. Figure 2-3 is divided into two portion (A & B) illustrating failure to danger and failure to safety of a system with components. The system components include bellows, spring, pivot and indicator. The indicator in portion "A" of Figure 2-3 indicates that the system has zero pressure, which is normally hazardous if undetected whilst portion B shows that the system is pressurised.

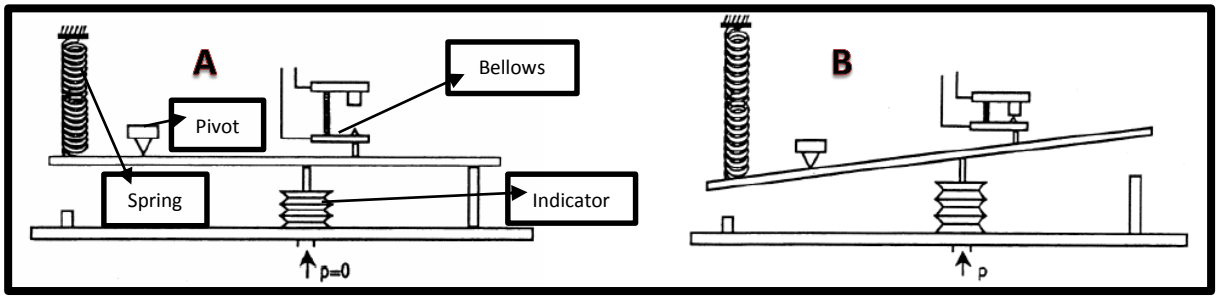


Figure 2-3-system for failure analysis of fail-safe and fail to danger conditions (Maheri 2013).

Furthermore, Figure 2-3-1 failed to safety after the system bellows broke down, indicator shows zero pressure whereas the system is pressurised, while Figure 2-3-2 failed to danger as

a result of fractured spring and indicator shows a pressurised system even when the system pressure is zero.

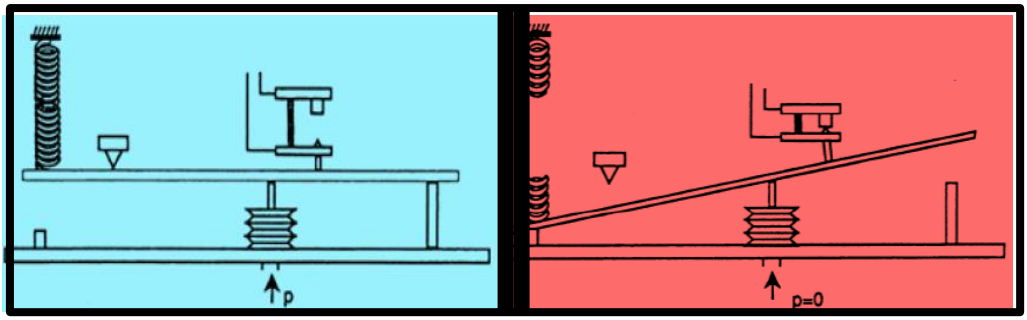


Figure 2-3-1 Failure to safety. Figure 2-3-2 Failure to danger (Maheri 2013)

Failure impacts on systems generally challenge reliability their position and can actually be prevented for better reliability (Keren *et al*., 2003), and also act as better risk management for a safety system (Pittiglio *et al*., 2014). Therefore, to prevent failure and make systems safer and reliable, the designers can also do the following:

- Redesign the system for "passive" improvement of reliability and safety.
- Identify possible points where the system component is most likely to compromise system operation, resulting in injuries and other losses.
- Carefully evaluate the effects of component failures on the system performance.
- Identify components that are critical to safety and
- Improve maintenance routines to reduce the likelihood of component failures.

2.5.3 Fundamental Measures for Reliability

Apart from FR, other ways of measuring reliability of a system include:

- Mean Time to Failure (*MTTF*)
- Mean Time Between Failures (*MTBF*)
- Maintenance Free Operating Period (*MFOP*)
- Probability of Success or Probability of Failure on Demand (*PFD*)

FR, MTTF and *MTBF* are time dependent reliability measures that are usually applied to systems that operate on a regular basis. FR clearly specifies number of failures on an hourly basis, the more failure is consistently specified, the less reliable such a system becomes, while the lesser the failure rate the more reliable the system becomes. Reliability is therefore increased with decrease in *FR.*

MTTF is the equivalent measure for non-repairable items, and can be referred to as the reciprocal of hazard rate (Paul & David, 2012). For *MTBF*, reliability is increased as *MTBF* increases and MTBF is usually specified in hours. However, *MTBF* can be impossible to predict if the time to failure distribution is not exponential (Dinesh Kumar *et al.*, 1999). In addition, MFOP currently represents one of the newest concepts designed to measure reliability particularly in the aerospace industry where dependence on *MTBF* can be reduced and specify reliability as a probability of time in service prior to failure (Dinesh Kumar *et al.*, 1999). One of the advantages is that it can track behaviour of a system throughout the life of the system.

PFD is mainly used for single-shot systems such as automobile airbags and fire extinguishers. It is specified as dimensionless probability or a percentage known as system safety engineering. These measures are mathematically modelled in (Lewis, 1996; Xie and Zheng, 2009) to account for detail knowledge of basic reliability calculations that conformed reliability practice as a function of time; for example, $R(t) = 1 - F(t)$.

2.6 System Availability Circle

Availability is a key component in reliability. It is regarded as a percentage of time that a system will be operationally viable to carry out its required function(s), which is mostly applicable to systems that operate continuously. It is mostly referred to as "five Nines availability", which represents (99.999%) of a system available state. Two important states that make up availability circle are failure and repairs. These states are in addition regarded as a prudent availability

concept. Based on the concept, a failure is followed by a repair and repair time normally puts system in an unavailable state as a result of down time.

There are different types of availability that further define system state and time. They include operational, intrinsic and steady state availabilities. First, operational availability takes into account administrative procedure times. Secondly, intrinsic availability excludes from consideration all other times in product or system lifecycle such as: accident management time, storage time, administrative or logistic time, and conforms to probability that a system can operate satisfactorily under given conditions (Houssin & Coulibaly, 2014). Finally, a steady state availability uses useful parameters to describe the amount of system available time (Wei et al., 2014). Steady state availability (Ass) and downtime (DT) are commonly used in practice in availability metrics. The mathematical expression of steady state availability is expressed in availability measures in Equation 2.8 overleaf, and is widely accepted in practice (Hairong & Jame, 2002). Interestingly, availability is one of the most fascinating characteristics of a repairable system (Maciejewski & Caban, 2008). It complements the fundamental basis for which system maintainability exists in the first place, since maintainability is a practice that enhances a system that is out of operation to be restored. In a nutshell, maintainability can predict how quickly and economically system operation can be restored so as to keep the system in an available state. With respect to these characteristics, system inherent availability(A_i) which is an important performance index for a repairable system, is usually estimated from the times-between-failures and the times-to-restore data (Wang & Dimitri, 2000). The estimation is then defined in Equation 2.7 which also reflects percentage of time a system would be available if delays due to maintenance, logistics and supply are ignored.

2.6.1 System Unavailability Estimation

A system can be unavailable due to latent failure that is yet to be detected, or it can be unavailable due to ongoing corrective and preventive maintenance of the system. So for non-

repairable systems or components, availability is equal to reliability. In other words, the probability that a system or component is available at time (t) is equal to the probability that it has not failed between 0 and t:

$$A(t) = R(t) \tag{2.6}$$

For a repairable system with ignored maintenance delayed time, the estimation of availability can be demonstrated as:

$$A_i = \frac{MTBF}{MTBF + MTTR} \times 100\% \tag{2.7}$$

However, in cases where the system never failed, the $MTBF$ becomes inestimable and A_i would be 100%.

In contrast, for systems that repair processes are not ignored, the steady state availability (Ass) is applied (Enrico Zio, 2007).

$$Ass = \frac{MTTF}{MTTF+MTTR} \tag{2.8}$$

Ass represents the system availability, $MTTF$ represents the mean time to failure, and $MTTR$ represents mean time to recovery (or repair). Also, Equation 2.9 shows that availability is a function of uptime and downtime.

$$A = \frac{UP\ Time}{Up\ Time + Down\ Time} \tag{2.9}$$

Uptime is the time in which the system is available for use, while downtime is the time in which the system is unavailable for use. The sum of uptime and downtime is equal to the total system run time.

Availability normally depends on reliability and maintainability, and operational availability is a function of active maintenance time as well as administrative delay and logistic delay time.

A concrete expression for operational availability is by the mean time between maintenance (*MTBM*) and the mean downtime (*MDT*) for each maintenance action. This similar expression was given in (Song *et al*., 2012) as:

$$A_o = \frac{MTBM}{MTBM + MDT} \tag{2.10}$$

Notably, the known established availability expression in Equation (2.8) does not hold 'per se' for redundant systems as failure rate under redundant systems are not constant but vary. However, for Equation (2.8) to hold validly for redundant systems, then $MTTF$ and $MTTR$ must be replaced by its equivalent, by ensuring that both are the inverse of the failure rate when failure rate is constant as shown below.

$$MTBF\ \&\ MTTF = \frac{1}{Constant\ Failure\ Rate} \tag{2.11}$$

To further simplify Up Time and Down Time from Equation (2.9), let T_p be the preventive maintenance time interval. This time can vary because an occurrence of a failure and the maintenance duration are random (Van der weide & Pandey, 2015), so for this purpose, several components can have different T_p (Tsai *et al.,* 2004). Let;

μ_m be the mean time for minimal repair of the component

μ be the mean time to repair of the component

$\lambda(t)$ be the hazard (failure rate) of the component

Then UP Time and Down Time can be derived using similar expressions in Equations 14 and 15 of (Tsai *et al*., 2004) where:

$$\text{Up Time} = T_p - \mu_m \int \lambda(t)dt \tag{2.12}$$

$$\text{Down Time} = \mu + \mu_m \int \lambda(t)dt \qquad (2.13)$$

Whilst extended reliability modelling insights, especially for the derivation of $\lambda(t)$, can be found in Maxim Finkelstein (2008), system unavailability is shown to be;

$$Downtime(DT) = (1 - Ass) * 8760 * 60 \quad \text{Downtime in minutes per year} \qquad (2.14)$$

To ascertain the usefulness of the explored measures above, many authors including (Ahmed *et al*., 2014; Cekyay & Ozekici, 2015) have done work using this simple and efficient approach for availability analysis. And more excellent availability measures comprising mathematical models of different systems are provided in (Chandrupatla, 2009; Sheng & Dhillon, 2011), proving that availability is estimated with full consideration of reliability and maintainability. In fact, by designing for appropriate achievement of availability and ensuring appropriate statistical calculations are entrenched, a very good maintenance programme can be achieved.

2.7 System Maintenance

Effective system maintenance is widely viewed as one of the major challenges confronting many industries and organisations, especially if the goal is to provide services at a higher competence level. Based on these challenging issues, many industries have placed a high priority on preventive maintenance to ensure system reliability level is satisfactory. Maintenance objectives are mainly achieved by staving off aging effects of wear, corrosion, fatigue, and related phenomena of such from systems through either one or both of the most popular maintenance types: preventive maintenance (*PM*) and corrective maintenance (*CM*). Preventive maintenance tends to avoid the failure occurrence of a system and reduces the potential failure consequences (Selvik & Aven, 2011; Lin *et al*., 2015). In this case, parts are replaced, lubricants are changed, or adjustments are made before failure occurs. On the other hand, corrective maintenance is performed after failure has occurred in order to return the system to service as quickly as possible. Both *PM* and *CM* have distinctive judgement criteria.

In PM, procedures is judged with regards to increasing reliability of the system, whilst the criterion for judging *CM* is mostly the systems level of availability discussed above, which is the probability that the system will be operational when needed. In both PM and *CM* reducing the huge amount of time and cost spent is usually a factor of concern. This is one of the reasons that the Reliability Centered Maintenance (*RCM*) concept has been adopted for more than two decades now to provide an alternative maintenance framework and expertise to address the time and cost gaps (Deshpande & Modak, 2002).

2.7.1 Reliability Centered Maintenance (RCM)

RCM was developed in the early 1960's by Nowlan and Heap as a well-structured, logical process for determining the optimum tactics for maintaining a certain piece of equipment, and to optimise maintenance procedures in the airline industry (Nowlan and Heap, 1978). The logic was subsequently adopted in the early 1980's to develop several aspects of space shuttle's maintenance. The logical process of *RCM* proved to be an efficient and effective approach for maintenance programs of all types. As part of *RCM* goals are to optimise maintenance and achieve the desired equipment reliability level at minimum cost, the practice also examined different possible ways that can lead to system failure and the appropriate maintenance tactics to manage failure (Bonnie & Donald, 2001). Particularly with the use of *RCM's* decision logic, the system analyst will endeavour to determine the best maintenance strategy for a specific failure mode. Simplicity is one of the strengths of *RCM*. It is usually not analytically difficult and rigorous if compared with other types of reliability analysis and can be accessible to both technicians and maintenance engineers. Decision making might however be very difficult considering questions without a certain answer (Eisinger & Rakowsky, 2001). In this case, safety is one of the troubling question gaps in traditional *RCM* application. An early question in the traditional decision logic concerns safety. Exemplifying this might be to verify if the system failure could have a direct, adverse effect on safety. Ideally such a simple question

deserves a yes or no answer, but in practice, gaps exist between the "yes" and "no" extremes. For every *RCM* analyst, the chances and choices of responding to the question differently for the same situation is not restricted, since no standard is offered by which to quantify the safety risk. The safety question can lead to significant error. Therefore, the essence of the question is to avoid unnecessary analysis in conditions where no safety risk exists. Furthermore, the choice of optimum maintenance strategy is normally performed with a decision diagram. For this reason, many different decision diagrams are proposed for use in *RCM* analysis. Amongst the decision diagrams is the one illustrated in Figure 2-4 as proposed by (Eisinger & Rakowsky, 2001).

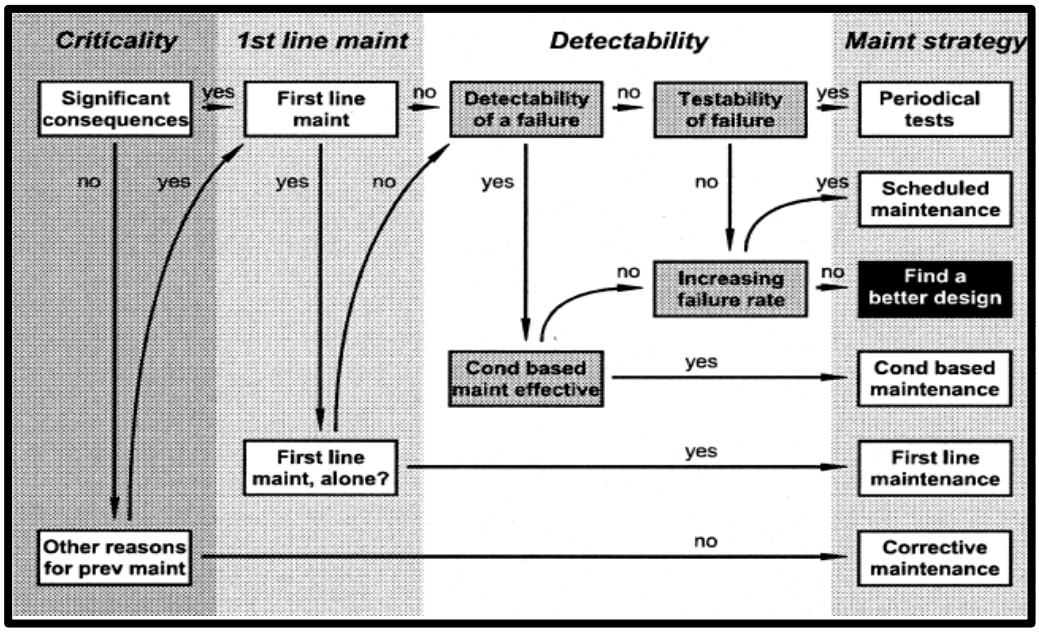


Figure 2-4 *RCM* Decision Diagram (Eisinger & Rakowsky, 2001)

From Figure 2-4, there are four main segments in the decision diagram namely; the Criticality, First line maintenance, Detectability and Maintenance Strategy Segments. Under the Criticality segment, the "significant consequences" are intended to account if the component breakdown implies reduction on system function. While "other reasons for preventive maintenance" tends to devise means for other possible maintenance strategies, hence in some cases preventive maintenance is more expensive than corrective maintenance, the second segment "First line

maintenance" is concerned to know if the operator is able to carry out online supervision and maintenance. Subsequently, the "First line maintenance alone" objective is to know if the first line maintenance is sufficient and effective. The third Segment "Detectability" is saddled with the physics of failure. Also, with the presence of "Condition-based maintenance effective" to ascertain if there exists a method for effective condition monitoring, in order to avoid component failure. The maintenance strategy finds the best maintenance strategy to be carried out. According to (Zhu *et al*., 2015 and Berdinyazov *et al*., 2009), maintenance of a system normally involves maintenance of multiple components with multiple failure modes, and each of which may require a different policy. This is because a maintenance policy maybe best and appropriate for one component and worst for another. In virtually all industries, maintenance policy is an inevitable reality, especially the two major types of maintenance: Preventive maintenance and corrective maintenance present in *RCM*. Their policies are quite distinctive and worthy of exploring.

2.7.2 Preventive Maintenance Policies

Preventive maintenance (*PM*) policy is a policy that improves system safety and reduces system failure rate. Usually the *PM* is triggered when the system reliability and availability falls below certain defined threshold, thereby making it an optimisation problem as to optimise weak parameters and minimise cost. The policy occurs when the system is still in operation with the aim of sustaining the system specific components in a certain condition (Liu *et al*., 2014). More recently, the policy has advanced from focusing on a single –component degrading system to a multi-component system due to increasing system complexity. *PM* is not only relevant in maintenance overview architecture as shown in Figure 2-5, it in addition has two leading classes in its policies to determine which component to maintain, and determine to what degree the component be maintained.

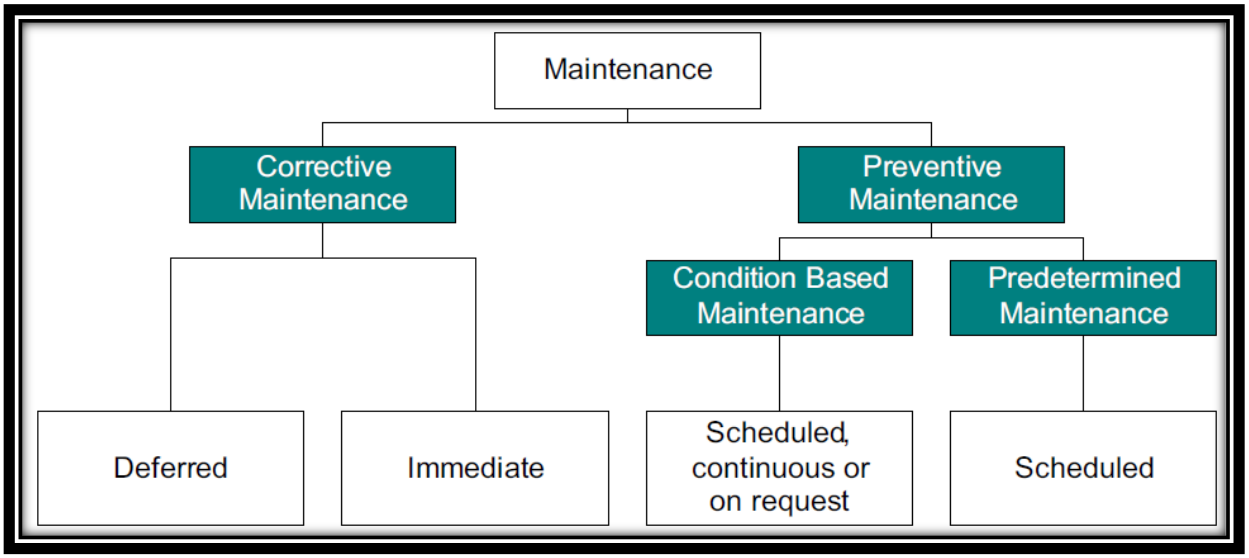


Figure 2-5 Overview of CM and PM Maintenance (Niu *et al*., 2010)

2.7.3 Significance of Preventive Maintenance Practice

Preventive maintenance is divided into two segments: Predetermined maintenance and condition based maintenance (*CBM*) as depicted in Figure 2-5. Niu *et al*., (2010) explained that a predetermined maintenance is a scheduled maintenance without the occurrence of any monitoring activities. For *CBM*, it monitors components and the entire system condition in order to ascertain a dynamic preventive schedule. With regards to maintenance scheduling, it may be based on the following:

- number of hours in use,
- number of times a component has been used,
- number of kilometres the components have covered and
- according to prescribed dates etc.

Preventive maintenance is usually very significant and has a very high preference over other types of maintenance. This can be due to the fact that it keeps safety checks alive and improves reliability and availability of a system through its policy and practice. It has two fundamental

policies: Either to implement PM after completing system assemblage or at each set-up time of the system component (Liu *et al.*, 2015). A carefully articulated PM policy and practice restores system components to as new as originally designed or somewhere close to as new. This development is however dependent on the following conditions:

- The extent of damage to, or wear and tear of the component
- The type and quality of technological tool used to effect the maintenance, and
- The level of technical knowledge or expertise gathered by the drafted maintenance personnel.

A well implemented *PM* saves cost (Mahmood & Maxim, 2014), while a poorly implemented *PM* increases cost (Phuc *et al*., 2015). Figure 2-6 provides a list of important benefits of performing *PM*.

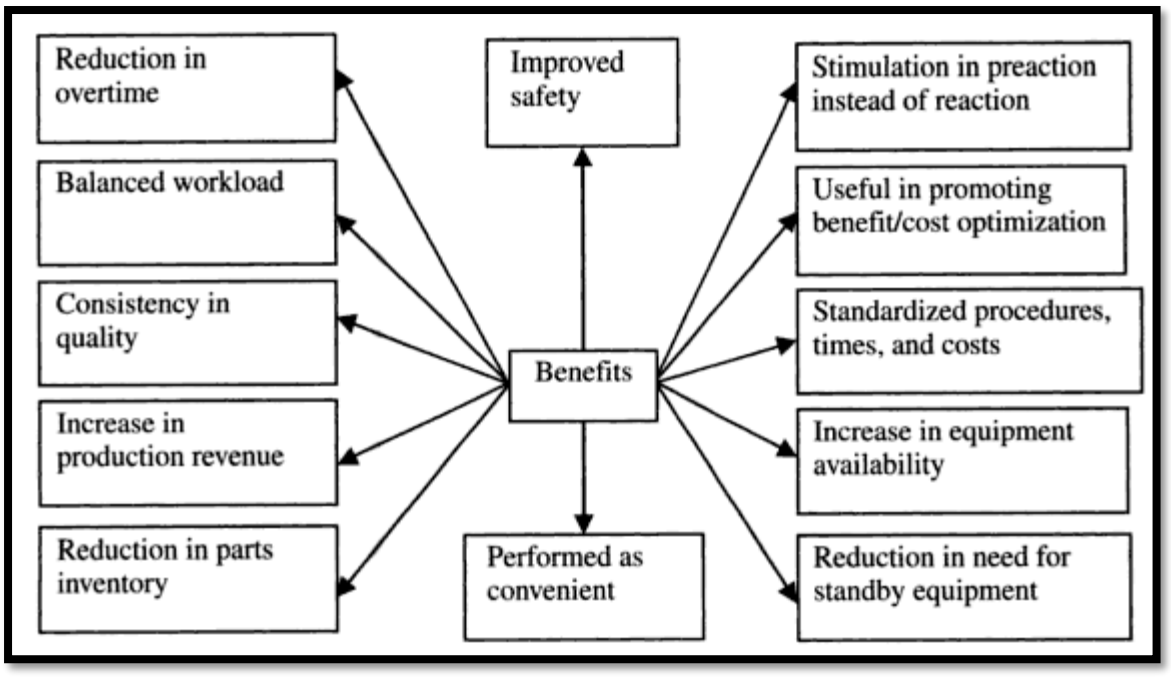


Figure 2-6– Benefits of performing preventive maintenance on a system (Dhillon, 2006)

2.7.4 Corrective Maintenance Policies

Corrective maintenance policy is a pure reactionary policy, where maintenance is carried out only when a system fails. In addition, the policy normally requires operating in an emergency mode with the aim of getting the equipment back in service as quickly as possible and in virtually new condition (Michael *et al*., 2000). CM is also referred to as repair (Liu and Huang, 2015). A repair activity can be carried out immediately or deferred, but it is usually targeted at making the malfunctioned or failed system to recover and be in "available state". These activities include a number of steps. Gao *et al*., (2012) showed that the steps include failure detection, isolation, decomposition, replacement, reassembling, testing etc. *CM* can be measured by "Mean Corrective Time" as defined by (Dhillon, 2006):

$$CMMT = \frac{\sum \lambda_i \ CMT_i}{\sum \lambda_i} \tag{2.15}$$

Where *CMMT* is the mean corrective time , λ_i is the failure rate of the i^{th} system component, and CMT_i is the corrective maintenance time of the i^{th} system component. *CM* policy is greatly flexible such that its effectiveness can be improved by adopting appropriate strategy. To improve *CM* effectiveness is by reducing its time, and this can be achieved through the following strategies (Dhillon, 2002 & Blachard *et al.*, 1995):

- Reduce accessibility time: a significant amount of time is usually spent accessing failed components. It is therefore imperative to recognise that careful attention to accessibility whilst design is underway can help to lower the accessibility time of components, and consequently, the *CM* time.
- Improvement of interchangeability: Effective functional and physical interchangeability is a fundamental factor during component replacement practice, thus it lowers *CM* time.

- Improvement of fault recognition, location and isolation: Experiences have shown that within a *CM* activity, Fault recognition, location and isolation consume the most time. Thus, good maintenance procedures, well-trained maintenance personnel, well-trained fault indicators and unambiguous fault isolation capacity are among the factors that help in reducing *CM* time.
- Consideration of human factors: During design, paying careful and detailed attention to human factors such as instructions, selection and placement of indicators; size, shape and component weight helps to lower CM time significantly.
- Employment of redundancy: This is mainly concerned with designing a redundant component with the capacity to switch in during the repair of faulty components such that the system continues to operate.

It is also to be added that other different maintenance policies in the literature are summarised in below Table 2.1.

Table 2.1- Summary of Maintenance Policies

MAINTENANCE POLICIES	**DEFINITION**	**SOURCE**
Predictive Maintenance	Is a policy adopted to save cost and replace system critical components before failure occurs, hence many system failures are quite dangerous	Giuseppe *et al*., 2010
Proactive Maintenance	It is a maintenance policy that prevents failure from occurring. It can detect, reduce or control the problems before complete failure occurs.	Zhong-Hua *et al*., 2013

Reactive Maintenance	Reactive Maintenance is also largely referred to as breakdown, repair, or fix-when-fail. It is a maintenance policy that assumes that failure is equally likely to occur in any system part or component, and that failure is age-related. Its policy precludes the identification of a specific group of repair parts as being more desirable than others. If the item fails and repair parts are not available, delays follow. If certain parts are needed to restore a critical machine or system to operation, a premium for expedited delivery must be paid. A complete reactive maintenance program ignores the many opportunities to influence equipment survivability.	NASA Facilities *RCM* Guide, 2008. pp 5-3.

2.7.5 Maintainability in Reliability

Maintainability constitutes design parameters that help in cost and time reduction (Atalag *et al*., 2014 & Barabadi *et al.*, 2011). It has four problematic (not been defined operationally viable) sub characteristics: analysability, changeability, stability and testability (Alf & Erik, 2009). These aforementioned sub characteristics however complement its efforts and qualities. Analysability entails having comprehensive understanding of system behaviour prior to making necessary changes. Changeability (i.e., a two –dimensional characteristics: relating to efforts expanded during changes implementation and the resulting quality of changes) is one of the avoidable critical maintainability features (Rippel *et al*., 2014). Changes are continuously introduced to match with new system requirements (Sun et al., 2014). Testability verifies and facilitates the establishment of test criteria (Sohn & Poong, 2006), and determines whether the criteria have been met through performance (Guanjun *et al*., 2014). Additionally, maintainability boosts system performance especially that of complex and dynamic systems.

From a literature review point of view, maintainability can often be overstretched to suit management preference. Since maintainability reduces the amount of maintenance time, management in a bid to overstretching maintainability tends to stress on maintenance (prevent or correct failure events) standard. That is by imposing certain additional maintenance requirements that are most times unnecessary in order to constraint maintainability and subject its achievements to depend entirely on management decision. The key maintainability figures of merit are the $MTTR$ (H. Garg, 2014), and a limit for the maximum repair time. While $MTTR$ includes access time, diagnosis time, spare parts supply, replacement time, checkout time and alignment time (Tsarouhas *et al*., 2009), the limit of a maximum repair time can be quantified. To quantify the repair time, let T be regarded as the continuous random variable signifying the time to repair a failed unit, with a probability density function of $h(t)$, then the cumulative distribution function $M(t)$ is computed below (Ebeling, 2001).

$$P(T < t) = M(t) = \int_0^t h(u)du \tag{2.16}$$

Equation (2.16) is a probability Equation certifying that a repair on the failed unit (u) will be accomplished within time t.

For Weilbull distribution, system maintainability is computed as;

$$M(t) = \Pr(T < t) = 1 - \exp^{-\left(\frac{t-\gamma}{\eta}\right)^{\beta}} \tag{2.17}$$

where (γ, β, η), are parameters that allow the computation of reliability.

γ – is a location parameter

β – is a shape parameter for describing the rate of change of failure rate

η - is scale parameter

For normal distribution, system maintainability is computed as;

$$M(t) = \Pr(T < t) = \Phi\left(\frac{v - t}{\sigma}\right) \quad (2.18)$$

where:

Φ Is the distribution function of a standard normal distribution.

v is the repair parameter of a normal distribution

t is the operating time.

σ is the standard deviation of the failure and repair rate of normal distribution.

For other distributions, the repair rate λ_r function is given as;

$$\lambda_r(t) = \frac{h(t)}{\sigma} \quad (2.19)$$

2.7.6 Repairs

Systems can either be repairable or non-repairable depending on the basic assumption on repair efficiency. Possible assumptions are minimal repair (As bad as old) and perfect repair (As good as new). In the minimal repair case, each repair leaves the system in the same state as it was prior to failure. Whilst the perfect repair case tries to fix the system and return it in a perfect and potential state that seems as if it was new. However, the reality between these peculiar cases is that: standard maintenance reduces failure intensity but will not leave the system as good as new (Doyen & Gaudoin, 2004). It is able to place the system state in a taxonomy that is better than minimal.

For major repairable systems, repair practice is carried out at the system failed state, or alternatively prior to failed state. In both cases, the system can either be repaired on-site or off-site. On-site repairs are less complex failure cases that can be corrected without keeping the system out of operation. Off-site repairs are complex (severe) failure cases that subject the system to be out of operation and taken to repair shop. Considering both cases, the steps required to be taken are not different, hence: the system is disassembled, evaluated and finally

repair is carried out to restore the system into a state in which it can perform its function again. An evaluation process determines the exact subsystem that has failed and needs to be fixed (i.e., if no redundancy) and also which non-failed subsystems shall be addressed under such circumstances. Besides these cases stated, many repairable systems consisting of one or more components and one repairman are deteriorative in nature due to accumulation of aging and damaging effects. Thus, ageing effects can be mitigated through performance of periodic preventive maintenance action to secure components rejuvenation. Such that some components are replaced in accordance with the hypothesis that ageing is a reasonable property of effective repair times. Moreover, effective repair activities increase the total system costs because it demands to have a sufficient number of repair technical team members on site at all times in readiness to intervene promptly when the need for the service arises.

This is normally followed by an increase in cost which is a preferable option, rather than reducing the number of technical repair team members on site or resorting to engage an external team which may lead to large production losses due to system downtime.

A well implemented repair on any system only restores the system working condition, but does not change the system age. For this purpose, system working condition paves way for two possible replacement policies. One of the policies is based on the system working age that is largely dependent on the system environment, and operational working conditions which affect component ageing process. The other is based on the number of system failures (Leung *et al*., 2011). According to (Zhang & Wang, 2009), implementation of these policies makes it feasible to observe that the more subsystems are replaced, the closer the system gets to its "new" (i.e., the original new state prior to use) state. Considering operational conditions, these conditions objectify the operational components modes (i.e., whether they are working continuously or not). The operational working conditions are as well related to the stress to which the component is submitted. Thus, in a more generic context, component operational conditions

could be expressed as a function of the number and duration of demands to operate for the ease of standby components, or through the operation time for components working continuously (Sebastian *et al*., 1999). While the environmental conditions objectify the environmental parameters under which the component is working, such as temperature and humidity. In safety related systems, some components work under different operational conditions. Over time, such components are placed in a very hard environment. For example, most times they are functioning under high temperature, while others remain in a very convenient environment. These factors usually have obvious negative effects on system inherent reliability; therefore, it becomes imperative that a system alongside with its components undergo repair to remain safe and reliable during operations.

Systems can attain reliability through redundancy allocation and through design diversity. Since redundancy allocation technique is a multi-objective problem, comprehensive discussion on the subject is given in the third chapter of this work. Design diversity is a crucial technique where component failure can be coped with (Zheng Wang *et al*., 2015). The technique also mitigates common-mode faults and can successfully be implemented in several different ways, which includes (Cristiano *et al*., 2015):

- Time or temporal diversity (i.e., repetition of the computation with different clock rates)
- Different hardware copies (hardware diversity)
- Employment of different design teams and design tools for the development of each system copy, particularly in a manner that commonalities are systematically avoided
- It can also be by using a different backup architecture approach (Ryouhei *et al*., 2010).

2.8 Techniques for Reliability and Safety System Analysis

To analyse a safety system is usually characterised with difficulty (Koo, 2005). This is mainly due to the complexity associated with such a system. They are modern and complex in nature

and may not fit in a conventional system analysis framework (Filippini & Andres, 2014). The complexity further subjects the engineers to consider all possible hazards so as to guarantee operational success. However, safety systems can either be analysed qualitatively or quantitatively depending on the perceptions of the system engineer. The qualitative safety analysis method comprises of a diagrammatic description of the factors that might cause accidents, while the quantitative safety analysis method simply estimates the probability of occurrence of each cause, which in turn can be used in estimating the risk of the accident (Netjasov and Janic, 2008). This can be represented pictorially. As shown in Figure 2-7, Pictorial representation of safety system analysis techniques drawn from Rouvroye & Van den Bliek (2002), clearly delineates which analytical approach falls within qualitative and quantitative. According to the author, it is appropriate to understand that quantitative analysis is mainly applicable where there is inadequate or no data on the system component.

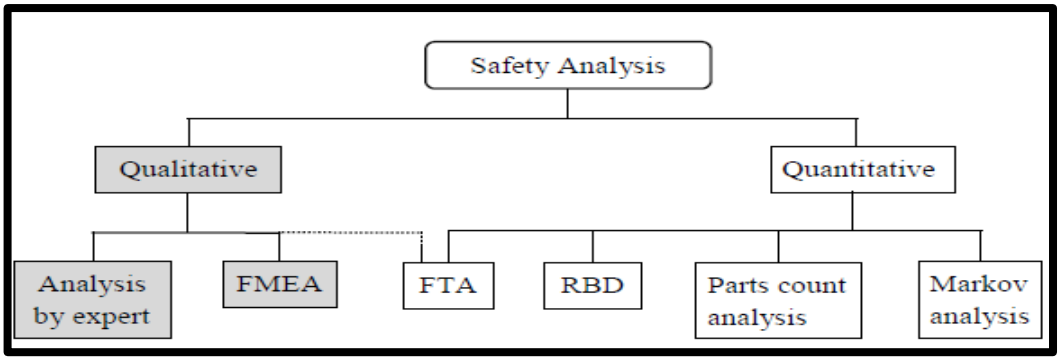


Figure 2-7-Techniques of safety system analysis (Rouvroye & Van den Bliek, 2002**)**

The procedure for each of the analysis method is defined in Hanea *et al.*, (2010); Yiannis *et al.*, (2015); Rouvroye & Van den Bliek, (2002) including Fault Tree Analysis (*FTA*), Reliability Block (*RBD*) and Double Failure Matrix (*DFM*). The *FTA* techniques analyses the failure mode of a system with special consideration on the possible events leading to such a failure mode.

2.8.1 Fault Tree Analysis (FTA)

William et al., (2002) concurred that *FTA* is simply an analytical technique where an undesired state of the system is specified (usually a state that is from a safety and reliability standpoint).

It is a deductive reasoning method because it identifies failures prior to their occurrence, analyses accidents, and acts as an investigatory tools to pinpoint failures. In contrast to deductive reasoning, *FTA* becomes an inductive reason method because it commences with general top event or output event and develops down through the branches to a specific input event that must occur so as to generate the output (Akgun *et al*., 2015). The faults in a system can be the events associated with mechanical failures, human errors, or any other related events with the tendency of leading to an undesired event. In clear terms, the sequence of events leading to probable occurrence of the undesired events is systematically divided into basic events whose failure probabilities can be estimated through logical interrelationships (Taheriyoun & Morandinejad, 2015). Quantitative and qualitative methods are apparent in *FTA* to ensure comprehensive reliability analysis of a system. The qualitative method deals with events and general factors affecting the system errors and excludes from consideration all numeric values, whilst the quantitative method is based on Boolean algebra where events either occur or not. However, the juxtaposition between these methods revealed that qualitative review is inherently required in the quantitative analysis. In other words, if failure data is assigned to the events, *FTA* measures safety through a risk based calculation or probability of failure distribution. If contrary (i.e., if no probabilities of failure data is assigned), *FTA* then falls under qualitative safety analysis. In practice, some of these techniques especially FTA and *FMEA* are generally static, and do not take into consideration the changes in system states, therefore are unable to capture dynamic system behaviour accurately (Yiannis *et al*., 2015).

FTA is constructed by using several symbols to represent various events and describes relationships. Amongst these symbols are: AND gate, OR gate, COMBINATION gate, *EXCLUSIVE OR gate*, *TRANSFER gate*, and *PRIORITY AND*. During construction, the concept does not consider all possible system failures or all possible causes responsible for failure of a system. Rather the concept is tailored to top event that corresponds to some

particular system failure mode, considering only those faults that contribute to this top event, and realistic by the analyst (Lavasani *et al*., 2015 and Boudali *et al*., 2007). Therefore, the logical output of the gates occurs or in other words, is applicable under the following fault scenarios: *AND gate* Output occurs if all of the input faults occur. *OR gate* output fault occurs if at least one of the input faults occurs. *COMBINATION gate* output fault occurs if n out of m of the input faults occur. *EXLUSIVE OR* gate output fault occurs if only one of the input faults occurs. TRANSFER gate transfers to/from another part of the fault tree, and *PRIORITY AND gate* output fault occurs if all of the input faults occur in a specific sequence. *FTA* construction steps are therefore provided as follows (CCPS, 2000):

- Recognise the system and the way it operates in order to clarify communications and interactions between system components.
- Identify top event as a unique event that represents a crisis state of the system capable of causing system failure.
- Construct fault tree diagram and reasonable development of failure (i.e., to logically connect causes of top event to each other using the various gates e.g. "*AND*" and "*OR*" *gates*).
- Qualitatively Analyse (i.e., to include ranking of basic events by determining their probabilities using data available and experts' ideas)
- Quantitatively analyse. In this case, to calculate the probability of top event.

Chen *et al*., (2014), and Piterka *et al*., (2014) separately employed *FTA* method in their individual works. Chen *et al*., (2014) employed the method to synthesize various damage modes of a composite airframe that suffers from complex damage during operations and figured out a variety of options addressing the causes. Piterka *et al*., (2014) used the method to analyse two systems: an emergency core cooling system and a containment spray system for a nuclear power plant with a *WWER 440/V213* reactor. Figure 2-8 depicts basic gates symbols

used in *FTA*, while a typical example of a constructed fault tree is showed in (Zhang *et al*., 2014). *FTA* is quite informative, and can as well be endless in finding causes, especially causes that the determination of failure probabilities involving human errors.

Symbol	Name	Description
	*AND Gate*	The output event occurs only when all the input events exist simultaneously.
	*INTERMEDIATE Event*	A fault event that results from interactions of other fault events that are developed through logic gates such as those defined above.
	*Basic Event*	A component failure that requires no further development. A basic event is the lowest level of resolution in a fault tree.
	*EXTERNAL Or HOUSE Event*	A condition or an event that is assumed to exist as a boundary condition for the fault tree.
	*TRANSFER Symbols*	The TRANSFER sysmbol indicates that the faultree is developed further on another page. The symbols are labeled using numbers or a code to ensure that they can be differentiated. Transfer symbole are often used to avoid repeating identical logic in several places in a fault tree model
	*OR Gate*	The output event occurs if any of the input events occur.

Figure 2-8 Description of boolean logic functions (Hyo *et al*., 2005)

2.8.2 Reliability Block Diagram (RBD)

RBD is a basic reliability analysis tool. It analyses systems, especially complex systems, in a manner that requires the complexity be broken down into the simplest form. However, the analysis is not capable of capturing repair or test activities (Verlinden *et al*., 2012). *RBD* of a

system actually represents the effect of component failures on system performance, and each component is represented by a black box that is assumed to be in one of two states: operating or failed (Forche, 1990). Most researchers in this context believe that the use of RBDs is one of the most essential steps for system reliability (Bistouni & Jahanshahi, 2014; Bourouni, 2013 & Levitin, 2007). For example, Yanjun & Wei, (2011) used the *RBD* approach to analyse system architecture and predict system reliability. Ding *et al*., (2014) proposed a novel method for safety integrity level (*SIL*) verification which plays a very critical role in reliability assessment of safety related systems, using RBD. Preference is made for the use of RBD in this work over others due to its ability to represent some of the redundancy strategies as shown in the Figures below.

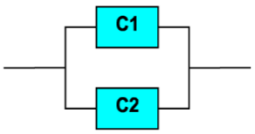


Figure 2-9 *RBD* of a simple active parallel redundancy

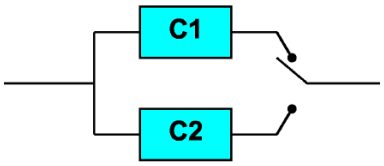


Figure 2-10 *RBD* of standby redundancy

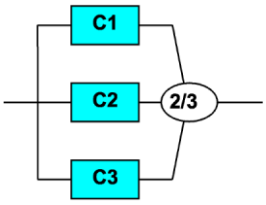


Figure 2-11 *RBD* of voting redundancy

Figure 2-9- fails only when both components (*C1 & C2*) have failed. Figure 2-10 also fails when all components have failed, while Figure: 2-11 fails when n out of m components have failed (in this case, 2 out of 3).

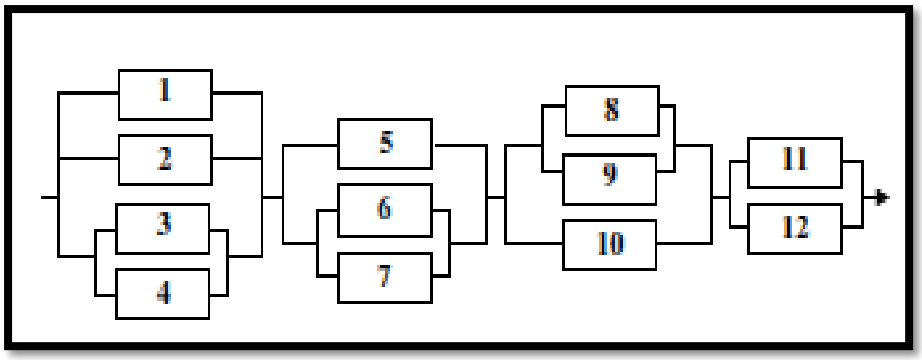


Figure 2-12 RBD of a data processing and transmission system (Levitin, 2007)

Figure 2-12 consists of four consecutive subsystems. Levitin, (2007) explained that the distributions of performances of components composing each subsystem are as follows; components $3\ and\ 4, 6\ and\ 7, 8\ and\ 9$ share the task and at the same time, compose two-component blocks. The rest of the components in each subsystem perform the same tasks in parallel with the two-component blocks (providing the task execution redundancy). The Figure used RBD approach to illustrate the importance of redundant component on any system as such component shares task to make a system reliable.

2.8.3 Double Failure Matrix (DFM)

DFM is an inductive technique that considers the effect of double failures. To explore its usefulness means the application will have to be limited to only noncomplex systems. DFM has various ways in which faults maybe categorised, this includes;

- Negligible faults,
- Marginal faults,
- Critical faults and
- Catastrophic faults (Roberts, 1987).

A major setback in the safety analysis approach is that the approach usually finds it difficult to analyse a system from its early design stage to completion due to the increasing scale and complexity surrounding the analysis approach. To address this setback, engineers and safety

experts now uses *HiP-HOPS* and *STAMP* techniques. Hierarchical Performed Hazard Origin and Propagation Studies (*HiP-HOPS*) and System- Theoretical Accident Models and Process (*STAMP*) are regarded as new classical safety system analysis techniques proposed to close the gap in safety system analysis, and are discussed below.

2.8.4 Stamp Based Analysis Techniques

Under STAMP-Based analysis techniques, each hazard analysis technique is based on a model of accident causation. The cause of an accident (i.e., an unplanned and undesired loss event involving human death, injury and other major losses including equipment etc.), is viewed as the result of a lack of constraints imposed on the systems design and operations (leveson, 2004). This satisfies the basic concept of the *STAMP* model that the most basic concept is not an event but constraints. That is the cause of accident, rather than being understood in connection to a series of events, is perceived as the result of a lack of enforcement of constraints imposed on the systems design and operation. Accidents only occur when system safety constraints are not enforced (Underwood & Patrick, 2014). Ouyang et al., (2010) proposed that the *STAMP* model of accident cause can be used to perform *STAMP*-Based Process Analysis (*STPA*). In other words, *STPA* is a safety analysis technique that is carefully handled based on the STAMP model of accident causation.

Additionally, the *STAMP* analysis technique views system failure in a manner that is contrary to traditional techniques. This can be mainly due to the criticism that traditional techniques depend largely on failure events. Traditional techniques do not also perform well in handling system accidents in the presence of dysfunctional interactions among operating components rather than failure of individual components (Changyong, 2014). The *STAMP* model is built on three basic concepts: Safety constraints, a hierarchical safety control and process models – along with basic systems theory concepts. These concepts attracted *STAMP* technique a widely distinguished recognition in the analysis of system safety. In *STAMP* analysis, system safety is

regarded as a control problem considering that enforcement of constraints are done by control loops between the various levels of the hierarchical control structure that are in place during system design and operations (Changyon, 2014). Based on basic loop process, a system accident is likely to occur due to dysfunction resulting not only from system component failure, but also due to incorrect control rules. The challenge in adopting this method is therefore, to ensure prompt and adequate identification of safety -related constraints, and impose them on the system at various levels. Levels in this case are both design and operational levels. In fact, *STAMP* appreciates dynamism in systems as it tolerates faults in case of any change due to the breach of a constraint, the system still operates safely. However, the goal of *STAMP* analysis is built on the three earlier mentioned basic concepts, while the new *STAMP* based system analysis techniques (*STPA*) goals identify accident scenarios that encompass the entire accidents process, including design errors, software flaws, complex human decision-making errors, and other factors contributing to accidents (Leveson, 2011).

STAMP-Based Process Analysis (*STPA*) technique is usually adopted at the systems early life cycle stage, and continues through the life of the system. During system design, it supports the realisation of a safety-driven process and helps shape early design decisions. At the early stage of system design, there is usually the possibility of a having shortage of useful information available to complicate design objectives and increase the challenges faced by the designer. In this case, adequate application of *SPTA* in hazard analysis becomes very handy in addressing information shortage issue leading to the challenges faced by the design engineer. The *SPTA* techniques achieves the analysis of hazard where design information is in short supply by first carrying out initial analysis that encourages flexibility and that can accommodate further emerging information while the design is underway. According to (Leveson & Dulac, 2005), "at such early design stage where information is little, the analysis of hazards using STPA will be very general at first and then will later be refined and augmented as additional information

emerges from the design activities". In accordance with the usefulness of this technique, several authors including Hata *et al*., (2015) have accomplished quality system designs that are safe (accident free) and reliable.

2.8.5 HiP – HOPS Techniques

The Hierarchical Performed Hazard Origin and Propagation Studies (*HiP – HOPS*) technique addresses the increasing complexity in design of modern engineering systems very effectively. The technique was aptly developed by Papdopoulos and McDermid in the late 1990s as a state-of –the-art compositional system dependability analysis tool capable of closing the design complexity gaps highlighted above. The central capability of this tool is the automatic synthesis of Fault Trees and Failure Mode Effects Analysis (*FMEAs*) by interpreting reusable specifications of component failure in the context of a system model (Yiannis *et al*., 2011). In Particular, because the analysis is largely automated, requiring only the initial component failure data to be provided; it therefore reduces the manual effort required to examine system safety. Hip-HOPS functions in conjunction with commonly-used system modelling tools, such as Matlab Simulink or Simulation X, (Sharvia & Yiannis, 2015). More recently, the concept has been extended to solve a design optimisation problem relating to reliability and cost through components selection and replication and alternative subsystem architectures (Yiannis *et al*., 2011). Hip –HOPS uses genetic algorithm (*GA*) to evolve initial non-optimal designs into new designs that better achieve reliability requirements quickly with minimal cost. That is, by selecting different component implementations with different reliability cost characteristics. Alternatively, it substitutes alternative subsystem architectures with more robust patterns of failure behaviour which is what one of the *Hip-HOPS* phases provides to explore many solutions from a large design space. In furtherance to *GA's* role, it exploits the automated fault tree and *FMEA* synthesis and analysis algorithms of a tool to calculate the fitness of each design candidate. *GA's* goal also identifies Pareto Optimal architectures for the

system which provides optimal trade-offs between reliability, cost and other parameters that may constitute the design objectives.

Hip-HOPS has three main phases: Model annotation, Fault tree synthesis, and Fault tree and *FMEA* analysis phase. The Model annotation phase keenly provides information to *HiP-HOPS* on how a component fails. The synthesis phase is the *Hip-HOPS* penultimate phase. What happens in this phase is the production of an interconnected network of fault trees that shows how component failures propagate from one component to another. In light of this, it relates how the component failure may progressively result to wider system failure. In the analysis phase which is perhaps the *Hip –HOPS* final phase, it analyses and examines the synthesized fault trees through automated algorithms to generate minimal cut sets of the system. Generated minimal cut sets not only describe necessary and sufficient combination of events that leads to undesired event, but also investigate the different scenarios to find for instance the redundant components that could be added to improve reliability (Mohamed-Larbi & Daoud, 2013). Quantitative analysis of the generated minimal cut sets data can then be used to evaluate the system reliability and unavailability values. In this case, the approach allows data combination into a multiple failure mode *FMEA* in order to show both direct and indirect effects of failure modes on the system. In general, *Hip-HOPS* encourages reusability by enabling failure-annotated components to be stored conveniently. Based on this, other components which are similar in type are allowed to reuse the failure data and save the designer the time and energy of having to re-enter the same failure data multiple times. Apart from analysis of the safety system to determine the reliability state of systems, another important issue in systems which is as important as reliability, is system redundancy. To properly handle redundancy, the two leading redundancy allocation strategies must be respected in practice. According to Kong *et al*., (2015), these strategies are active and standby and are further classified as hardware redundancy as shown in the Figure 2-13.

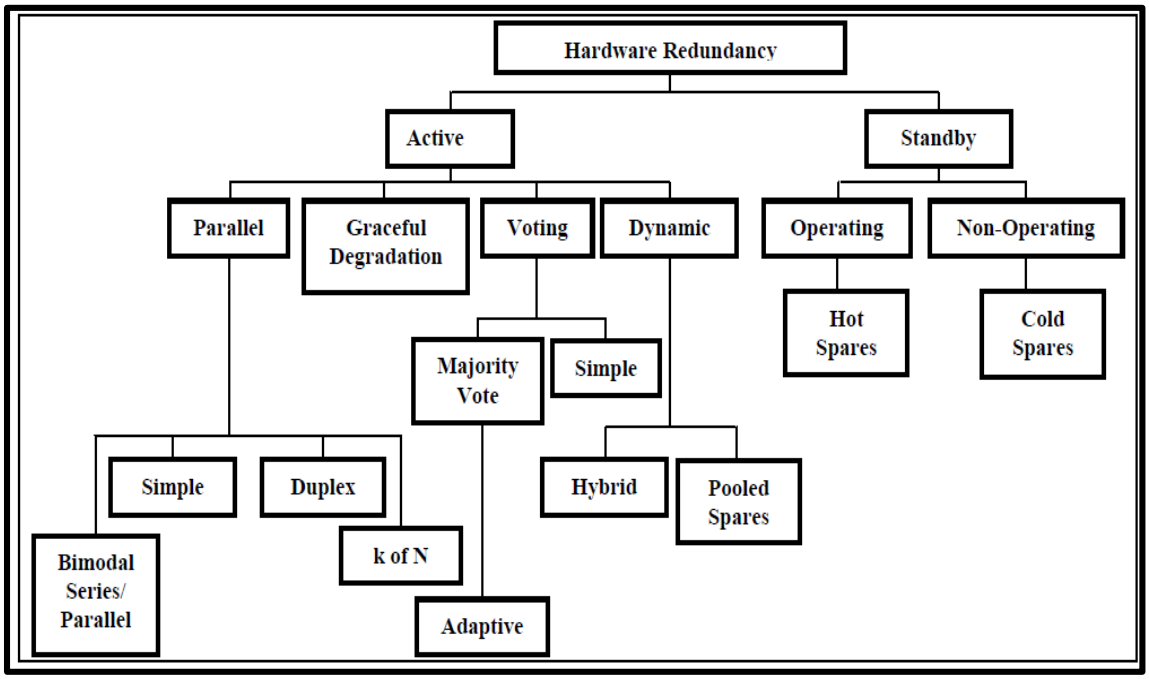


Figure 2-13 Redundancy strategies (Technical Manual 5-698-1, 2007)

2.9 Active Redundancy Strategy

Ardakan and Hamadani (2014) explored significantly the best period to engage in active redundancy strategy. In general, all redundant components units in active redundancy simultaneously start operation from time zero, even though only one component is required in a particular time. It does not require an external component to perform a detection function, decision and switching when a component or system path fails. They automatically pick up the load for a failed unit. A typical example of this system is multi engine aircraft, where the aircraft can continue to fly with one or more engines out of operation. The principal component forms a parallel system, and the redundant component undergoes the regular stress level as the principal component. Parts can be replaced, and system performance can as well be improved by introducing this kind of redundancy strategy (Misra & Misra, 2011). In this case, active redundancy may be suitably adopted to provide spares to the coherent system either as component redundancy or system redundancy. More excellent insights about active

redundancy including their application have been explored in Bueno & Carmo, (2007); Valdes & Zequeira, (2006).

2.9.1 Standby Redundancy Strategy

In the standby redundancy arrangement, the redundant components are used sequentially in the system at failure times. Apart from that, the strategy is further compartmentalised into three variants known as; cold, warm and hot (Nanda & Hazra, 2013), and is usually employed when system replacement takes a negligible amount of time without causing failure on the system. When a component is effectively carrying out the required function within a specific time without failure, that component is referred to as the active one. While the other component(s) that back-up, is the spare or standby. The spare can switch to be active only when failure has occurred on the active component (Zhao & Liu, 2005). The failure idea of a standby redundancy is that the system fails when all the components have failed, and when a component failure is detected as soon as it occurs, it is usually possible to use standby redundancy. The three standby strategies (cold, warm and hot) depend on whether the system process of ending is tolerable or not (Soltani *et al.,* 2013). For Cold standby, it implies that the redundant units, when not required are not in use and thus have zero failure rates. For warm-standby, it portends a possibility for the redundant components that failure rates of components are lower than hot-standby redundancy, while hot-standby redundancy is so close to active redundancy and all components might fail (Yahyatabar & Jahromi, 2012). Excellent insights including a failure rate model of the aforementioned redundancy strategies are explored in Lewis, (1996) & Zoulfaghari *et al*., (2014).

2.10 Methods of System Configurations

Reliability of a system depends on a few fundamental principles. First, the quality of the design materials, the designer's level of knowledge and the configuration (arrangement) methods adopted during the system assembly. Often a system can be literally found having many

components (i.e., a multi component system). In this situation, the system reliability becomes the function of the individual components reliability, and the adopted configuration method. A System configuration can take either of the two known fundamental ways: Series configuration or parallel (redundant) configuration. Each of these can be demonstrated by means of a reliability block diagram (RBD) as shown below.

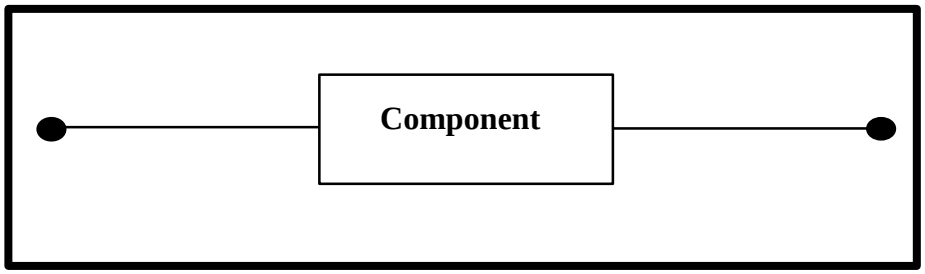


Figure 2-14 Simple reliability block diagram

A multi-component system can also be represented with end points having a network of blocks. In such connection, the system is regarded as a working system if the end points connection between the source and the sink nodes exist (i.e. x and y), otherwise it is regarded as a system in a failed state if the endpoints connection does not exist. To show systems with M , the same number of components in series and parallel configurations using a reliability block diagram (*RBD*) is demonstrated in Figures 2-15 and 2-16 respectively.

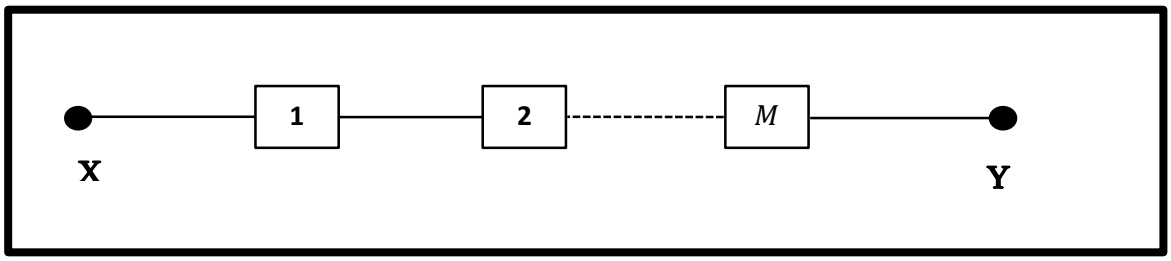


Figure 2-15 Reliability block diagram for a system in series

Figure 2-15 is a typical simple series system configuration. The system is made up *M*, independent modules consisting of components 1 and 2. All the module with components must operate in order for the system to function properly. Components 1 and 2 can have a specific

range of component choices to make, and it absolutely depends on the total components M, available.

A series system fails completely if either of the components $(1\ or\ 2)$ fails, and this development makes it less reliable when compared to a parallel system configuration. Should each component be represented by A_i, such that $1 \leq i \leq M$, series system reliability can then be represented as a probability that all the components are in a working state. That is, $R_s =$ Probability (All components are working), and the mathematical expression is shown in Equation(2.20) .

$$R_s = P(A_1 \cap A_2 \cap, \dots, A_M) \qquad 2.20$$

(PA_i) is the probability that component A_i, is in a working condition.

Reliability is very vital, especially in relevant industries where systems are expected to successfully carry out operations. Given this significance, it is therefore expedient to increase series system reliability through any best possible way or strategy. One of the ways that has been proposed in the literature by Ahmadizar & Soltanpanah, (2011), to increase system reliability is through the addition of components to each module in parallel. This strategy increases the system reliability and reduces system failure rate. With regards to Parallel system configuration, the system is said to be in a failed condition if all M, components found in the system have failed. Applying the same A_i, components notation to represent the parallel system configuration, reliability will therefore assume this order; $R_s =$ Probability (All components are failed). Mathematically,

$$R_s = P(A_1 \cup A_2 \cup, \dots, A_M) \qquad (2.21)$$

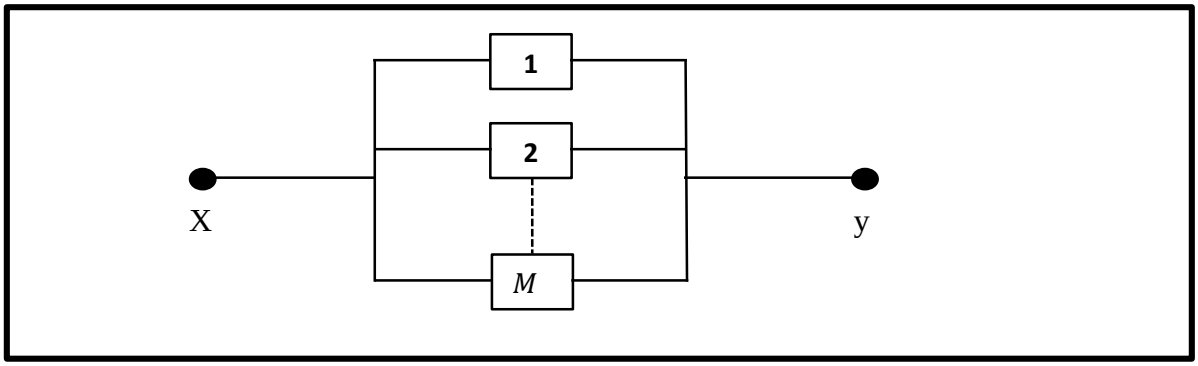


Figure 2-16 Reliability block diagram of a system in parallel

Figure 2-16 is a simple parallel system configuration designed to function even under a failure scenario of either of its subsystem. The description of the subsystem/module is in similitude with that of series configuration. The system has a total components M, and two independent components 1 and 2 in each of the modules. Components 1 and 2 can have a specific range of component choices to make which absolutely depends on component availability. However, the significant difference here is that the subsystems are connected in parallel. A parallel system connection is one where the success of any one of the components in the system results in the system overall success. Failure of a single or more subsystem does not lead to system failure, instead the system only losses a portion of its productivity. This is because alternative parts will continue to provide operation. For this reason the parallel system configuration outperforms that of series and proves to be more reliable.

Using these configuration principles, system configuration can extend to series-parallel and parallel-series through appropriate combination of both system structures. A typical Series-parallel system structure is depicted overleaf in Figure 2-17. Such structure is usually a coherent system made up of series modules consisting of parallel components (Nahas *et al*., 2007). On the other hand, a Parallel- Series system is a coherent system that can be describe as a parallel arrangement of disjointed series modules, see Figure 2-18.

2.10.1 Series-Parallel System Configuration Method

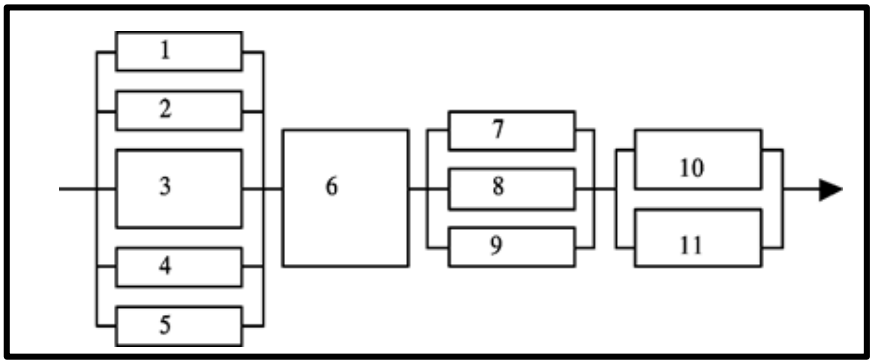


Figure 2-17 Reliability block diagram of a Series- Parallel System (Bris *et al.*, 2003)

Conceptually, series-parallel system configurations consist of subsystems connected in series, such that each subsystem can have several elements connected in parallel as shown in Figure 2-17. The capacity and productivity of the system elements is fundamental, especially with respect to overall system performance evaluation. Components can fail and be repaired to make the system available, because availability targets to satisfy customers demand. Additionally, components contained in subsystems are characterised according to their availability, cost and performance (Ouzineb *et al*., 2008). From Figure 2-17, the entire system structure contains four series modules consisting of eleven parallel components overall. Each of the components is assumed to be in good working condition and is reliable.

2.10.2 Parallel-Series System Configuration Method

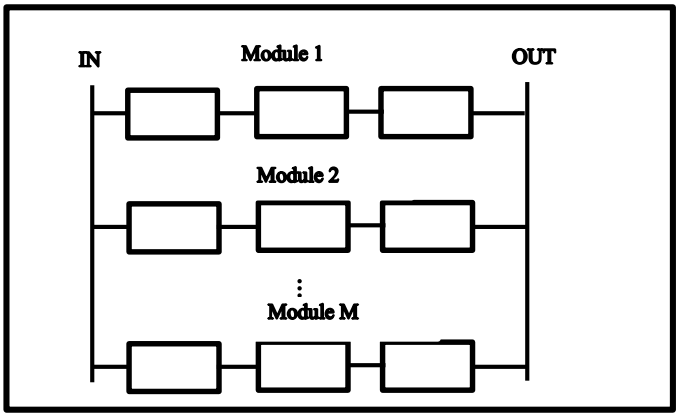


Figure 2-18 Reliability block diagram of parallel -series (Liu,2003)

Under a Parallel- Series system configuration, modules are connected in parallel as shown in Figure 2-18. The system also has a very good reliability and performance level. In addition, a typical optimisation problem concerned on the best way to determine the optimal number of parallel components in each subsystem, is usually a reliability constraint. A reliability constraint imposes a minimum requirement of subsystems/system reliability.

From reliability literature, there are, practically, many other types of system configurations, among them are $'k-out-of\ M'$ and complex systems. A $'k-out-of\ M'$ system is similar to a parallel system, as long as at least $k,$ channels function, the system will function (Lu *et al.*, 2008 & David Coit *et al*., 2015). For a complex system, it neither takes the shape of a series nor parallel system in the arrangement of its structural components. Rather, many numbers of component and links that aid interactions between the system and environment are usually present.

2.10.3 Complex System Configuration Method

A complex system is characterised by large numbers of components, cut sets or link sets, or by statistical dependence between the components states (Kiureghian & Song, 2008). These characteristics make it considerably difficult to understand and design. The first characteristic is that it interacts within the system and the environment in order to avert possible failure. The second is that correlations exist between the system components and its environment to restrain the system from getting more complex. In other words, it controls the dynamics of complex systems. In the case of designing a complex system, the parts can either be identical or non-identical. To this end, it is appropriate to refer to Boland & El-Neweihi, (1995), whose work extensively addresses how to determine appropriate system configuration conditions that can tolerate identical or non-identical parts. Moreover, complex systems make it quite challenging

to realise the exact system objective during the design process and system prediction time. This complexity emanates as a result of such system behaviour which steadily increases in size, dependencies and interactions amongst subsystems (Huang *et al.*, 2014). A typical example of a complex system is depicted in Figure 2-19, whilst additional useful insights concerning complex systems can be explored in (Constantinou *et al.*, 2015).

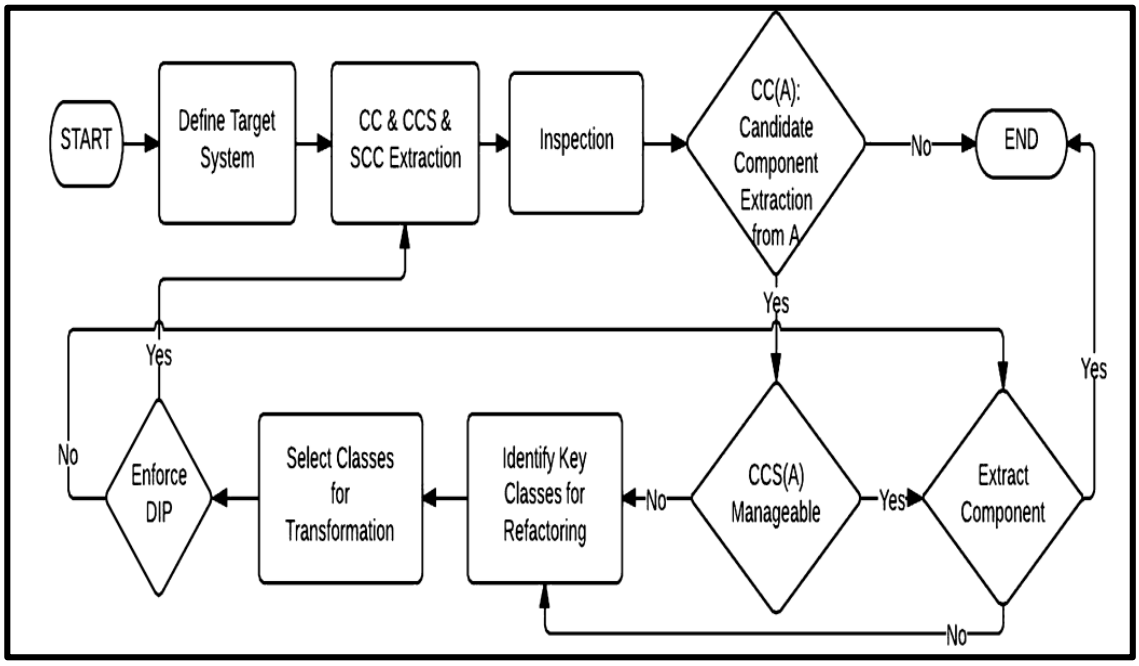


Figure 2-19 Complex system structure (Constantinou *et al.*, 2015).

To ascertain reliability of a complex system can be actualised through several analytical methods such as Fault Tree Based method, Matrix Multiplication method, Decomposition method, Minimal Cut-Set methods and many more. A Comprehensive overview of these methods has been explored by the following authors: Volkanovski *et al.*, (2009); Song & Won-Hee, (2009); Yeh *et al.*, (2015); Yevkin, (2009) and Rahman, (2011). However, for most complex systems with multi-level such as Mechanical, Hybrid Renewable energy etc., applying some of these methods can be difficult and time consuming. Based on this, several other methods have been further proposed by other researchers to cushion the gaps found in the literature. Amongst these methods is the bounding method for full-system reliabilities based

on sub-system tests, without requiring subsystem independence Stacy *et al*., (2013), and a simulation modelling method for complex system reliability proposed by Cao *et al*., (2012), as a more technical approach to realizing complex system reliability.

Apart from that, another fascinating type of solution for actualising reliability of a complex system is the Monte-Carlo (*MC*) Simulation technique. Monte-Carlos is a noteworthy computational tool for simulation, the idea is to simulate a complex system (or part of it) and determine the system performance for easy decision making (Canale *et al*., 2014). The method has been widely adopted by many researchers in the recent years to address complex system reliability due to its appropriateness to complex systems. The only unfriendly side of the MC method is the computational efforts that may be involved (Naess *et al*., 2009). Thus, the computational time does not grow abruptly as in the case of analytical methods. With the latest advancement in computer technology, faster processing machines that are capable of optimising the runtime of the simulation algorithm are now accessible. Many publications in this area have been made, and they are generally detailed on the use of simulation methods using different approaches. Amongst these approaches is the Hull girder reliability assessment approach to provide accurate estimates for the failure probability with reduced computational cost. In addition to that, Haifeng & Asgarpoor, (2011) proposed a sequential Monte Carlo simulation based method to generate reliability and maintainability history charts for which the reliability indices, as well as their probability distribution were successfully calculated. System reliability is practically about reducing failure rate and possible estimation of time to failure, T , which is a continuous random variable. The uncertainty associated with T can be described using an appropriate cumulative probability distribution function of system failures $F(t)$ which is characterised by the probability $P(T \leq t)$, as defined earlier in equation(2.3).

2.11 Probability Distributions for Modelling Time to Failure

There are many probability distributions for modelling descriptive characteristics of the continuous random variable T. However, it is quite important to determine the distribution that best describes the failure pattern because the choice of distribution can adversely affect a calculated reliability value of a system, with respect to some distributions associated with constant failure rate. The two main leading probability distributions are exponential distribution and weilbull distribution. The exponential distribution is mostly applicable to electronic components and complex systems, while the weilbull distribution is mostly applicable to Mechanical components.

2.11.1 Exponential Distribution of Failure Pattern

Exponential distribution is one of the very important types of distribution. It is often also referred to as negative distribution in reliability literature, and perhaps the most widely applied statistical distribution in reliability field. One of the reasons for its importance is that the exponential distribution has a constant failure rate function (Artur, 2013), while for other distributions, the rate at which failures occur varies with time. With regards to other distributions, failure rate cannot be discussed, instead the term Hazard Function (i.e., a function that describes how the rate of failures varies over time) is used (Wang *et al*., 2002). This type of distribution is used to model assumptions of a constant failure rate or hazard rate, λ, which is an instantaneous rate of failure ($\lambda > 0$) . Implying that the probability that a system or component that has survived time t, will fail within a limited interval of time,$'t, t + \Delta t'$ is constant. The failure probability $\lambda \Delta t$, is independent of system age and with respect to modelling a system life, it can be shown to have the exponential distribution (negative) using the assumption of constant failure rate. The probability distribution is defined with respect to equation 2.2 which has a constant failure rate as:

$$F(t) = 1 - exp^{-\lambda t} \quad (2.22)$$

where:

$F(t)$, is the probability density function (*PDF*) over time t .

t, is the length of time the system must function

exp, is the base of natural logarithms

λ is failure rate (inverse of *MTBF*)

This Equation is applicable only when the rate at which failures occur is constant.

Furthermore, the above relationship in Equation (2.22) can also be shown for the reliability function $R(t)$, generated in Equation(2.1), and the density function of the time to failure $f(t)$, in Equation(2.5).

For Reliability function $R(t)$, the expression becomes:

$$R(t) = exp^{-\lambda t} \quad (2.23)$$

Where:

$'R(t)$ ' is the reliability over time t.

While for the density function of the time to failure $f(t)$, the new expression is shown in the equation below.

$$f(t) = \lambda exp^{-\lambda t} \quad (2.24)$$

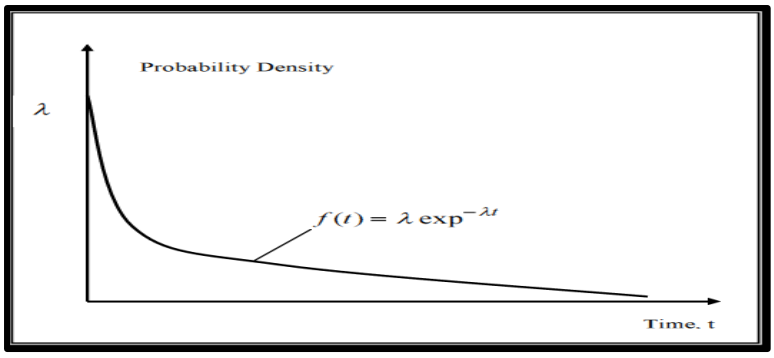


Figure 2-20 Probability density function of negative exponential distribution (Todinov, 2005)

Further mathematics involving special cases in this kind of distribution such as "The two-parameter exponentiated exponential (*EE*) or generalized exponential (*GE*)" where failure rate function with other cases like mean residual lifetime and many more are carefully explored by David Han, 2015 and Nandini *et al*., (2010).

2.11.2 Weilbull Distribution

Weilbull distribution is mainly applicable to mechanical systems and can be used to manage moderately large amounts of data. It was introduced by Professor Waloddi Weilbull in the early 1950's and became so popular for modelling phenomenon with monotonic failure rates. It is widely used in reliability engineering and elsewhere due to its versatility and relative simplicity. It requires a large amount of experience to be able to use it in practice. Three parameters without physical meaning are fundamental in Weilbull distribution. These parameters are (γ, β, η), and they are parameters which allow the computation of reliability and MTBF. However, one unpopular feature of the Weilbull distribution is that it fails to provide a good fit to data sets with bathtub shaped failure rates that are often encountered in reliability literature (Singla *et al*., 2012). The cumulative Weilbull distribution is given as:

$$F(t) = 1 - exp^{-\left(\frac{t-\gamma}{\eta}\right)^{\beta}} \qquad (2.25)$$

where:

β, is the shape parameter, also known as the Weilbull slope. It clearly describes the rate of change of failure rate, increasing or decreasing.

η, is the scale parameter

γ, is the location parameter

To model a variety of life behaviours depends on the values of the shape β, and scale η, parameters. These values affect the distribution characteristics, and a change in the parameters value can transform the distribution into an exponential or normal distribution. The value of the location parameter γ is usually zero (Smith, 2001). So when the value of β=1, and the value of $\eta = 1/\lambda$, the Weilbull distribution takes the exponential distribution shape as depicted in Equation(2.22). Further transformation takes place when the value of $\beta > 3$, in this case, the distribution is approximately normal. Now, using the reliability relationship initially established in Equation(2.1), to show that the above cumulative Weilbull distribution can also be shown for the system reliability or survival function $R(t)$.

$$R(t) = \exp^{-(t/\eta)^{\beta}} \qquad (2.26)$$

Extended mathematical models with relevant reliability concepts are detailed clearly in Todinov, (2005).

2.11.3 Other Existing Distributions

Apart from the two famous distributions aptly summarised above, due to their effectiveness in estimating industrial failure probability, there are other useful distributions that registered their presence in the literature. Amongst them are Lognormal, Uniform, Binomial and Poisson. First, the Binomial distribution, otherwise known as a discrete distribution (i.e., a distribution that allows a random variable to take only two values, say '0' or '1' with equal probability) provides statistical independence in reliability analysis, especially at component level. In fact, it defines the ideal state at which a system component will work. The binomial distribution approach entails that the probability that a component will be working is completely independent of the state (working or failed) of other components. Furthermore, the probability of success in each trial (the probability that the component will be working) is the same. This is mathematically

demonstrated below, supposing that X, is the number of failures in n independent trials, R is the probability of success at each individual trial, then X, is a random variable with Binomial distribution:

$$P(X = r) = \binom{n}{r} R^{n-1}(1-R)^r, \quad 0 < R < 1, \qquad r = 0,1,\dots,n \qquad (2.27)$$

According to Ming Han, (2012), R, is also called the reliability of product at the censored time, t . Generalised mathematical Binomial distribution models and their functions are carefully handled in Bergeron, (2013).

The Poisson distribution is also discrete in nature, but quite different from the binomial distribution. Both distributions can be carefully adopted for the modelling of defects. While the binomial distribution models the number of defects or incidents in a fixed sample size with a fixed failure probability, the Poisson distribution models the occurrence of some phenomenon such as the amount of defects within a fixed region of space using lambda (λ) as its parameter. The probability distribution function of a Poisson distribution is given as;

$$f(x) = \frac{\lambda^x \exp\{-\lambda\}}{x!}, \; x = 0,1,2,\dots,n \qquad (2.28)$$

where:

λ is the number of occurrences

x is a random variable.

These distributions usually have different graphic shapes, and typical examples of lognormal, normal, Poisson and binomial distributions are shown in Figure 2-21 overleaf.

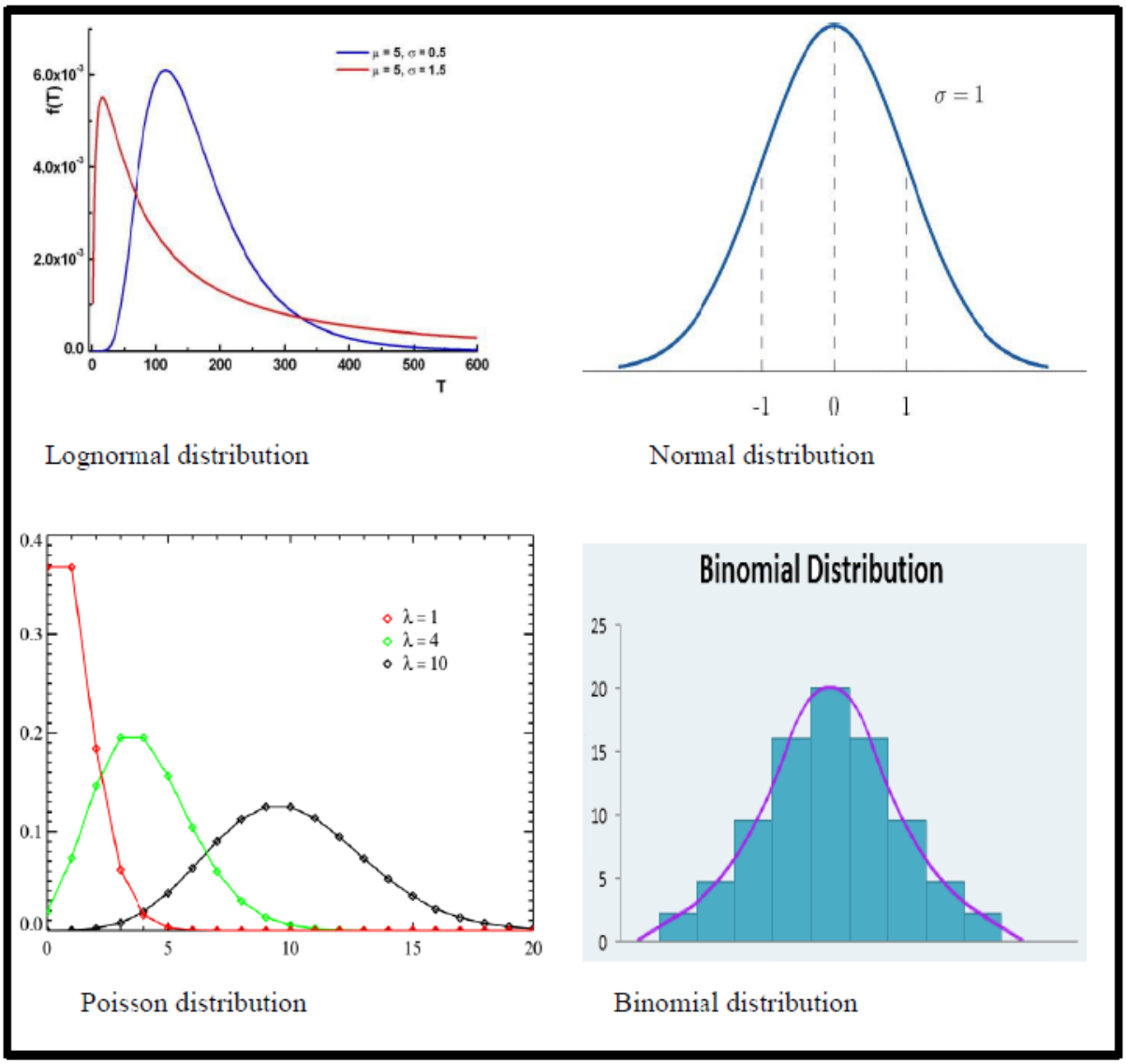


Figure 2-21: Schematic of distributions function (Erik *et al*., 2012).

Therefore, to know the exact distribution to use in data management can be found in Nikolaidis *et al*., (2005). The author summarised in Table 2.2 that to apply each distribution method largely depends on data size.

Table 2.2 - Selection Process for All Classes of Distributions (Nikolaidis *et al*., 2005)

Distribution	**Typical Uses for Distribution**
Normal	▪ Large amount of data. ▪ Large amount of experience.

Uniform	▪ Small amount of data. ▪ Good quantification of physical bounds. ▪ Large amount of uncertainty of most likely values.
Beta	▪ Small amount of data. ▪ Good quantification of physical bounds. ▪ Good quantification of most likely values.
Weibull	▪ Moderate to large amounts of data. ▪ Large amount of experience.
Lognormal	▪ Moderate to large amounts of data. ▪ Large amount of experience.
Poisson	▪ Models total number of occurrences of some phenomenon during a fixed time period or within a fixed region of space.
Binomial	▪ Models number of defects or incidents in a fixed sample size with a fixed failure probability.
Hyper distribution	▪ Small amount of data. ▪ Poor quantification of physical bounds. ▪ Large amount of uncertainty of most likely values.

2.12 Design for Reliability

Design for reliability is crucial as it starts in the idea phase of the product development cycle and continues through the product obsolescence. It is employed to effect positive product reliability improvement by utilizing physics-of-failure knowledge to design out potential problems. Very often, it is related with two other blocks (Reliability verification and Analytical Physics) of activities to constitute a coherent design process. With regards to reliability verification, it targets to satisfy customers' reliability objectives, which are usually two fold: design maturity testing and process reliability. Given these developments, design maturity

testing demonstrates that customer's needs will be met when the product is eventually exposed to demanding conditions, while process reliability provides a basic foundation for achievement of a realistic accelerated design maturity test. The second activity which is analytical physics, gathers knowledge about a system's physics-of-failure; which is concerned with understanding how and why systems fail. In addition, during reliability design, different design stages are accounted for, starting with the conceptual stage and continuing through to the final system stage. These processes include the following: Idea, Evaluation, Development, Transition and Production stage to ensure product reliability, see Figure 2-22.

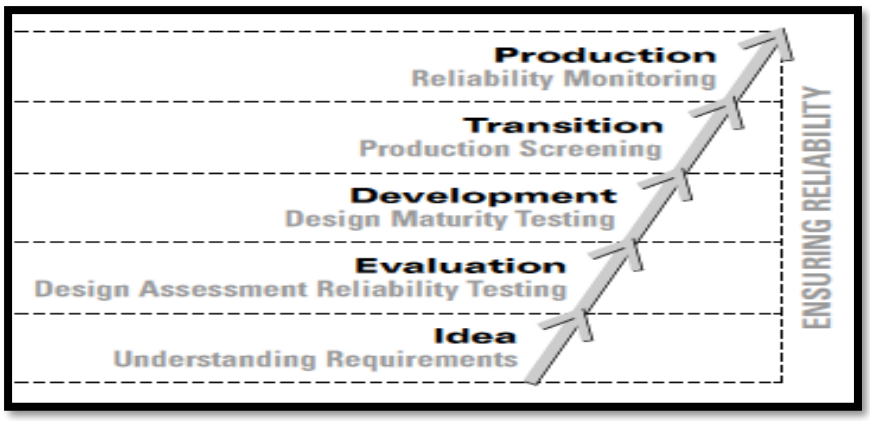


Figure 2-22 Design for reliability process (Crowe, 2000)

Firstly, the idea phase applies concurrent engineering processes for reliability design. That is to have an exact understanding of the customers' requirements, which often includes the tools of Failure Modes and Effects Analysis (*FMEA*), Product competitive benchmarking, and reliability predictive modelling which is used to direct the design approach. Deployment of these tools is used to reduce associated design risks in order to record success when a product is eventually lunched into market as a prime target. To this end, the first real reliability impact occurs at this stage, and the idea phase is the first concept where a design solution is conceived to define the ultimate reliability level that can be guaranteed, see Figure 2-23.

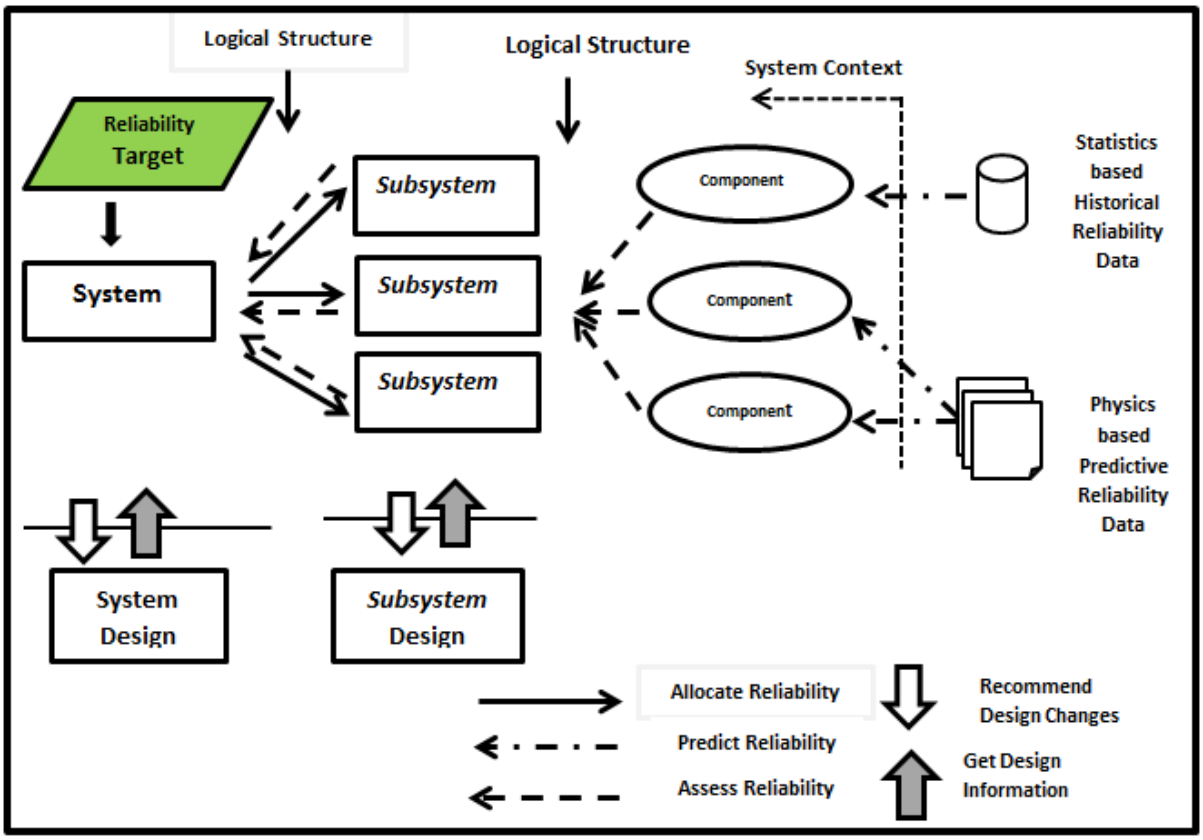


Figure 2-23 Conceptual reliability design model (Injoong, 2010).

From the above Figure, dash-dot lines show how a component reliability can be determined from a statistics-based model or a physics-based model. System reliability is then assessed from components to subsystem to parent system. When the reliability assessed is higher than or equal to the assigned target reliability, it implies no design modification is required or recommended. However, when the assessed reliability is less than the assigned reliability target, the design changes are recommended.

In a situation where system designs are not possible nor available, design changes may be initiated at the parent system level, so that redundant/modularized/alternative subsystems or other approaches can be pursued.

Secondly, the evaluation phase accounts for the design risk and its possible mitigation mechanism because design-risk mitigation activities occur at this phase. Evaluation goal, therefore, is to resolve possible uncertainties around the chosen design approach through

performing risk-mitigation studies to ascertain the level of feasible reliability growth and improvement. Under this activity, reliability growth worth 65 percent can be achieved from the initial design point before proceeding to the reliability development phase for proper design validation. This phase validates and demonstrates that the reliability design is capable of meeting its targets. In this case, the target is to know if the expected operating-life requirements identified from the design conceptual stage have been met, using Design Maturity Testing (*DMT*). *DMT* is normally based on physics of failure and afterwards, gives way for the transition phase to be effected on the design.

Under the transition phase, the design is carefully screened to ensure robustness in early production units, as well as check for infant mortality problems. That is, to assist in defining the proper screening method capable of preventing infant mortality failures from occurring. This process is usually uneasy as adequate failure mechanisms must be found and employed to carefully detect flaws and retain a product's useful life. Lastly, the production phase defines a proper reliability monitoring process and technique to control design performance and ensure it is in steady state over its life time. Reliability monitoring adequately prevents process variation of a product cycle from affecting product reliability.

Whilst engineering design is important because it is reliability driven, in many cases the cost of the system is also a very important objective to consider. Cost and reliability are normally two conflicting objectives. Quite often, a design is driven by reliability as the main objective with other assessment criteria, including the cost, as the secondary objectives. The designer saves cost by designing for reliability and dealing with uncertainties. Uncertainties are known phenomenon that usually confront design engineers, their occurrence on a system can lead to huge financial loss. They also occur in loading, material properties, geometry and other aspects of functional systems. In most cases they are viewed in different taxonomy as reducible or irreducible uncertainties. Reducible uncertainties are caused by a lack of data, modelling

specifications, human errors, etc. while irreducible uncertainties happen at random, and no known measure has been found capable of reducing their occurrence. To this end, handling a great deal of uncertainties in modelling and evaluation allows designers to carefully consider one or more of the following:

- Collecting more data
- Understanding the problem better
- More quality control, especially in terms of using more reliable components instead of using standard components,
- Implementing redundancies in the system,
- Incorporating diversity to avoid common mode failure,
- Alternating the design concept, for example, towards reducing the number of components or to remove the critical components from the design, and
- Adding fail-safe mechanisms to control the consequence of a failure.

Additionally, designers introduce deterministic design approach by using safety factors to cover the effect of uncertainties on system performance. The only shortcoming with a deterministic design approach is its inability to predict reliability performance of a system ahead of time. According to Nikolaidis *et al.*, (2005), designers must have a complete understanding of the system, subsystem and its components performance under nominal and off-nominal conditions. Such that when a deterministic design approach is applied to various scenarios, the consequences if any can be evaluated. Based on this, designers mainly consider one of the following:

- Robust design, in which the functionality and performance is less sensitive to the variation of design parameters, and
- Designers also adopt a non-deterministic design in which the uncertainties in modelling are quantified and taken into account in the early design stage.

However, application of deterministic and nondeterministic design approaches can assist in meeting the goals of a successful design only if the designer rightly understands the design sensitivities and validities. Sensitivity refers to the likelihood that an effect will be detected, while validity refers to the likelihood that; what is detected is in fact, of interest. In other words, as regards to a deterministic design approach, the designer takes into account the existence of uncertainties and considers them in the design, through application of suitable safety factors. The assumption is that, careful introduction of safety factors into the design and considering worst- case scenario, leads to design of a reliable system.

2.12.1 Deterministic Design

Under deterministic design, key principles based on deterministic considerations have served as the backbone of typical real life systems designed for safety. Amongst these principles include; defense-in-depth, safety margin, redundancy, diversity, and independence. These principles are absolutely useful design concepts, and as a result, sustain their important roles in keeping the aforementioned kind of systems safe. This is because an ideal system designed for safety analysis, primarily focuses on events initiation rather than event consequences. Which means a step-by-step sequence of events from initiation to final stabilized conditions should be addressed for each initiating event. However, preponderant emphasis is still placed on initiating events, as no systematic method capable of identifying the associated event consequences is provided (Ahn *et al*., 2010). This also constitutes one of the deterministic design approaches weaknesses. Apart from that, a couple of performance deficiencies of typical real life systems are as inherent with this approach. Take for instance, Hybrid Renewable Energy Systems (*HRES*) where Maheri (2014), highlighted the following difficulties during a critical evaluation of a deterministic design approach of a standalone *HRES*:

- Accurately predicting the cost of the system
- Does not evaluate power reliability of the system directly, and

- Its solutions lead to unpredictable power reliability
- In fact, no reliability measure is calculated as part of design candidate assessment.

These weaknesses can be overcome through replacing the traditional deterministic approach for making the design and decisions with a balanced risk-based one that uses rigorous models to quantify uncertainties and assess safety. In short, this is the exclusive role of the non-deterministic approach.

2.12.2 Non-Deterministic Design

Non-deterministic design approaches help in designing safer and cheaper systems than a traditional deterministic approach. As mentioned earlier, the approach accounts for uncertainties in the operating environment, the material properties and the accuracy of predictive models. Uncertainties may include the following;

- Variability (natural uncertainty): Uncertainty due to inherent randomness or unpredictability of the physical system; irreducible and can only be quantified in a statistical sense.
- Model parameter uncertainty (data uncertainty): Incomplete knowledge of model parameters/inputs due to insufficient or inaccurate data; reducible by sufficient data or accurate measurements.
- Model structure uncertainty (model uncertainty): uncertain model formulation due to approximations and simplifications in a model; reducible by improving model formation (Agarwal, 2004).

In general, engineering design problems mostly assume complex forms especially when they have more than one design objective (multi-objective). In this case design objectives become conflicting in terms of finding the best objective that will stand out as the best design solution against the other objectives. This scenario leaves system designers to figure out the best adoptable approach to arrive at the best solution without compromising other objectives.

Certainly, improvement in one objective leads to a worse solution for at least one other objective. One of the realistic approaches is the use of multi-objective optimisation. Multi-objective optimisation is a realistic way of solving many complex engineering problems. The approach can resolve conflicting design objectives through optimisation as detailed in chapter three of this work.

2.13 Chapter Summary

In this chapter, the author revealed and analysed various parameters used to measure reliability and extensively discussed reliability concepts which clearly showed that the concept is a function of time. Reliability of a system decreases with time due to several factors such as component aging, failure and the use of substandard design materials. These factors remain a concern in the engineering field which needs to be addressed. To address these factors, the author has established that a good design approach capable of addressing the physics of failures at every design stage should be adopted. In the author's view, there are various ways a system can fail, and often the cause of failure originates from system faults which subsequently result in system failure if not promptly detected and fixed.

The chapter subsequently demonstrated that paying much attention on system maintenance and maintainability are fundamental for reliability improvement of systems. In addition, several maintenance practices capable of addressing both system components that can successfully undergo maintenance and the ones that cannot were explored. Furthermore, the condition and necessity leading to incorporation of design for system reliability were handled alongside with types of system configurations and modern analysis techniques by which a system can be effectively analysed. From the explored literature, it revealed that multi-objective optimisation approach is more a realistic solution approach amongst other approaches used in solving

complex engineering problems. Therefore, the following chapter will extensively explore various multi-objective optimisation methods and techniques of solving complex engineering problems. Afterwards, multi-objective approach will be adopted in chapter four to solve multi-level formulation problem as one of the complex engineering problems addressed in this work.

3. Redundancy Allocation: A Multi-Objective Optimisation Problem

As stated earlier, redundancy allocation problem (*RAP*) is one of the most widely applied techniques to increase system reliability, Safari, (2012); Pourdarvish & Zahra, (2013). It is effort driven, requires more resources and involves selection of components with appropriate levels of redundancy. *RAP* help to reduce a system *FR* which also results in a reliability increase of a system through appropriate optimisation process. An appropriate optimisation process as proposed by Ran & Chi-Ming, (2012) mostly targets to increase reliability as the best objective. Several other authors including Valian *et al*., (2013); Garg *et al*., (2014) & Sahoo *et al*., (2012) have published works buttressing the relevance of reliability optimisation, and have no doubt that reliability can consistently be improved upon by formulating and addressing *RAP*. A proper *RAP* formulation can therefore be obtained traditionally to handle all systems redundancy levels.

3.1 Traditional Formulation of Redundancy Allocation Problem

Traditional RAP formulation handles multi-level redundancy and can be formulated as a general multi-objective optimisation problem to cope with redundancy at multi-level. This is mainly because, RAP is regarded as a multi-objective optimisation problem (Zai, 2009). Multi-Objective Optimisation Problem (MOOP), usually optimise objectives simultaneously. As a matter of fact, objectives can conflict such that preference for the best solution is unknown. For a such scenario, decision makers' trade-off conflicting objectives and express preference over another. Many authors including Ouyang *et al*. (2015) have formulated MOOP using

different design parameters, but according to Marler & Arora (2004), general MOOPs can be best formulated as follows:

$$\underset{\mathrm{x}}{\text{Maximise}} \quad \mathbf{F}(\mathbf{x}) = [F_1(\mathrm{x}), F_2(\mathrm{x}), \ldots, F_k(\mathrm{x})]^T \qquad 3.1$$

$$\text{Subject to } g_j(\mathrm{x}) \leq 0, \quad \mathrm{j} = 1,2,\ldots,m$$

$$\text{where } l = 1,2,\ldots,e, \quad h_{(X)} = 0$$

Where k is referred to as the number of objective functions, m is the number of inequality constraints, and e is the number of equality constraints. $\mathrm{x} \in E^n$ is vector of design variables (in other words, also known as *decision variables*), where n is the number of independent variables x_i. $F(x) \in E^k$ is a vector of objective function $F_i(x): E^n \rightarrow E^1$. $F_i(x)$ are also referred to as objectives, criteria, payoff functions, cost functions, or value functions. The gradient of F_i (x) with respect to x is written as $\nabla_{\mathrm{x}} F_i(\mathrm{x}) \in E^n$. $\mathrm{x}_i{*}$ is the point that minimises the objective function F_i (x). Any comparison $(\leq, \geq, \text{etc.})$ between the vectors applies to corresponding vector components. The feasible design space **X** (often called the feasible decision space or constraint set) is defined as the set $\{\mathrm{x} | g_j(\mathrm{x}) \leq 0, j = 1,2,\ldots,m \text{ and } h_i(\mathrm{x}) = 0, \ l = 1,2,\ldots,e\}$. The feasible solution $\boldsymbol{Z}$ (also called cost space or the attainable set) is defined as the set $\{F(\mathrm{x}) | \mathrm{x} \in X\}$. Feasible criterion space and attainable set are both used in the literature to describe $\boldsymbol{Z}$. There is an indirect distinction between the ideas of feasibility and attainability. While feasibility implies that no constraint is violated, attainability implies that a point in the criterion space maps to a point in the design space. Each point in the design space maps to a point in the criterion space, but the reverse may not be true; every point in the criterion space does not necessarily correspond to a single point $\mathrm{x} \in$ **X.** Consequently, even with an unconstrained problem, only certain points in the criterion space are attainable to realise design objectives.

Actually, objectives changes and can either be for example to maximise cost whilst minimising weight of a system, and maximising performance whilst minimising the system energy consumption rate. As a result of this, it is unlikely that there exists a single best solution. One of the ways to handle such a situation can either be to transform the problem into a single objective problem by combining all design objectives to form a single aggregate objective function. Alternatively, to employ a weighting system to construct the overall function as applicable in the current practice.

Weighting systems currently comprise a set of weighting factors and turning exponents. Turning exponents are referred to as the degree/level of importance attached to a single objective with respect to other objectives. Weighting factors represent the relative importance degrees of objectives having significant influence on the location of the final solution on the Pareto frontier. The fundamental goal of a multi-objective optimisation is to identify solutions in the Pareto optimal set and analyse the trade-off between conflicting objectives (Kitayama & Yamazaki, 2012). These solutions could be classified as non-dominated, Pareto optimal, Pareto efficient or non-inferior, if none of the objective functions can be improved in value without degrading some of the other objective values. However, identifying the entire Pareto optimal set for many multi-objective problems is practically difficult due to size. Furthermore, many problems especially combinatorial optimisation problems, to proof solution optimality is computationally infeasible. Based on this, a practical approach to multi-objective optimisation is the investigation of a set of solutions (the best known Pareto set) that represent the Pareto optimal set as well as possible (Konak *et al*., 2006). Considering these, a multi objective optimisation should have the following qualities:

- The quality of ensuring that the best- known Pareto front is able to capture the whole spectrum of the Pareto front. This measure requires investigation of solutions at the extreme of the objective function space.

- The quality of distributing solutions in the best-known Pareto front uniformly, and diverse over the Pareto front in order to provide the decision- maker a true picture of trade-offs.
- The quality of positioning the best-known Pareto front to be as close as possible to the true Pareto front. The reason is because in an ideal sense, the best- known Pareto set should be a subset of the Pareto optimal set.

3.2 Pareto Optimality

According to Marler Arora, (2004), under Pareto optimal;

$A\ point,\ \ x^* \in X, is\ a\ Pareto\ optimal\ iff\ there\ exists\ no\ other\ point, x \in\ X, such\ that$

$F(x)\ \leq F(x^*), and\ F(x)\ < F_i(x^*)\ for\ at\ least\ one\ function$.

All Pareto optimal points lie on the boundary of feasible criterion space, but often there exists algorithms that provide solutions that may not be Pareto optimal but may satisfy other criteria. For example, weakly Pareto optimal; a point is said to be weakly Pareto optimal *iff* there is no other point that improves all the objective functions simultaneously. For example, a point, $x^* \in X$, is weakly Pareto optimal *iff* there exists no other point, such that $x \in\ X$, such that $F(x) < F(x^*)$ as shown in Figure 3-1

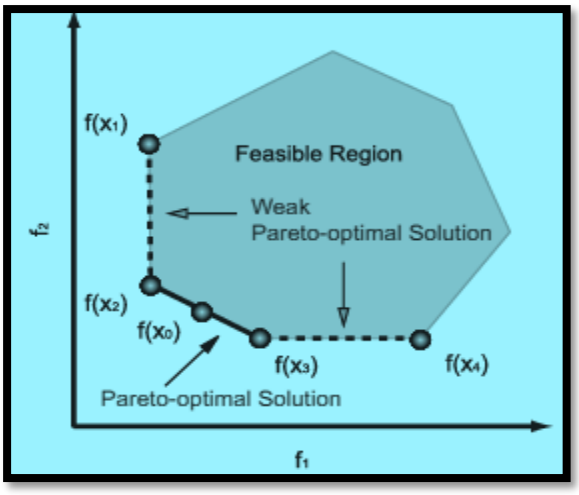


Figure 3-1 Pareto optimal solution and feasible region (Hiroyasu *et al*., 2007)

Figure 3-1 shows Pareto optimal solutions with respect to two objectives. The solid line shows the optimum solutions, while the dotted line shows the weak optimum solutions. From the Figure 3-1, no other point is capable of improving all the objective functions simultaneously. However, all Pareto optimal maybe categorised as either proper or improper. For proper Pareto optimal: A point, $\mathrm{x}^* \in X$ is said to be a proper Pareto optimal if it is already a Pareto optimal and there exists some real number $M > 0$ such that for each $F_i(\mathrm{x})$ and each $\mathrm{x} \in \mathrm{X}$ satisfying $F_i(\mathrm{x}) < F_i(\mathrm{x}^*)$, there exists at least one $F_j(\mathrm{x})$ such that $F_j(\mathrm{x}^*) < F_j(\mathrm{x})$ and $\frac{F_i(\mathrm{x}^*)-F_i(\mathrm{x})}{F_j(\mathrm{x})-F_j(\mathrm{x}^*)} \leq M$ (Geoffrion, 1968). The quotient is referred to as trade-off, and it represents the increment in the objective function j resulting from a decrement in objective function i. Geoffrion, (1968) and Ghosh, (2014) maintained that if a Pareto optimal point falls short of the above condition enumerated, then it is said to be improper. That is, a point $\mathrm{x}^* \in X$ is called a weak or improper Pareto optimal point if there is no other point $\mathrm{x} \in X$ such that $F(x) \leq F(x^*)$.

3.3 Methods of Solving Multi-Objectives Optimisation Problems

The mathematics of the multi-objective decision model is not new. Many research works have been carried out by different authors to advance their area of interest by adopting different suitable multi-objective optimisation methods. There are different methods of solving multi-objective optimisation problems. For some methods, transforming the original problem into a sequence of single objective optimisation problems is necessary. A transformed multi-objective optimisation problem into a sequence of single objectives, leads to solving numerical problems involving a nonlinear optimisation technique. This method remarkably allows the designer to have great flexibility during decision making. In either method preference is given, involves good computational effort in numerical determination of the Pareto optimal set. For the purpose of clarity, preference is an intrinsic function (i.e. points in the criterion space) that exist in the mind of the decision –maker, which perfectly incorporates his/her preferences. For example, selecting a set of weights that reflects preference towards one objective or another can be difficult, and preferences tend to be discrete. Secondly, on the manner in which the transformations exercise of an original problem into a sequence of single objective optimisation is implemented. There are several other methods of multi-objective optimisation solutions as discussed in the below subsections.

3.3.1 Weighted Sum Method

Under this type of method, the problem is transformed to a single objective by combining all design objectives and forming a single aggregate function. In other words, the objective function (U) becomes a weighted sum of the objective functions of the multiple objective decision model as exemplified below.

$$U = \sum_{i=1}^{k} w_i \, F_i(x) \qquad (3.2)$$

where $w_i \geq 0 \; and \; \{w_i\} \neq 0$, " if any one of the weights is zero, there is a potential for the solution to be only weakly Pareto Optimal" (Marler & Arora, 2010).

This model is advantageous due to its simplicity. By altering the weighting factors $\{w_i\}$, a set of Pareto solutions can be generated. In order words, the weighting factors $\{w_i\}$ reflect the relative importance of various objective functions, and it is straight- forward to implement the algorithm. The weighting factors also have significant influence on the location of the final solution on the Pareto frontier due to the direction of search. Normally, search is directed towards the points closer to endpoints corresponding to the objectives that have been considered more important. First, the Pareto optimal points corresponding to the boundaries of solution space are obtained by carrying out separate optimisation on each objective function. Followed by a systematic alteration of weighting factors $\{w_i\}$ until a sufficient number of Pareto optimal solutions are found. Naidu *et al*., (2014) successfully applied this method to optimise a controller's parameter of a Load Frequency Control (LFC) of an Electrical Power System. The only shortcoming about this method is that, it can be difficult to distinguish between setting weights to compensate for differences in objective-function magnitudes and setting weights to indicate the relative importance of an objective.

3.3.2 Decomposition-Based Method

This method employs a scalarizing function in order to carryout conventional optimisation. By this method, the multi –objective problem is reduced to a set of single objective problems for the purpose of solving it easily. In this context, "easy" simply implies that the problems assume a straightforward shape in which upon the solution yields a good approximation of the set of optimal solutions (Giagkiozis & Fleming, 2015). A typical example of this method was adopted

by Zhang & Li, (2007), they both successfully used decomposition based strategy to decompose the multi-objective optimisation problem into a number of scalar optimisation sub-problems, and simultaneously optimised them. The method proved to be very good in terms of performance. According to Zhang & Li, (2007), apart from the appreciable performance levels, each sub-problem was optimised using only information from several neighbouring sub-problems to reduce computational complexity at each generation.

3.3.3 Converting Objectives to Constraints Method

This conversion method advocates for all-but-one design objectives to be treated as constraints. The multi-objective optimisation problem is transformed to a single objective and is prone to conflict the design objectives. Therefore, in event of conflicting objectives, the solution obtained by this approach is a single point on the Pareto frontier of the original problem. The reason is because the designer consciously imposed direct constraints on the locus of the solution prior to start of the optimisation (Lin, 1976). For example, supposing a design for minimisation of (x) was anticipated, where (x) is any suitable objective(s), the problem formulation to convert (x) to constraints is demonstrated below:

$$min\ f(x)$$
$$x \in \Re^n ; f \in \Re^m$$
$$subject\ to\ g_i(x) = 0$$
$$h_j(x) \leq 0$$

$\rightarrow$

$$\min f_d\ (x)$$
$$x \in \Re^n$$
$$subject\ to\ g_i(x) = 0$$
$$h_j(x) \leq 0$$
$$g'_k\ (x) = 0$$
$$h'_l\ (x) \leq 0$$

$$x = \{x_1, x_2, \dots, x_n\},\ \ f = \{f_1, f_2, \dots, f_m\} \qquad (3.3)$$

Successful practice of this method was implemented in electrical power and energy systems (Zhang & Liu, 2008). These authors converted the multi-objective problem into a single-objective problem using the fuzzy optimisation technique. The fuzzy optimisation technique idea is to simultaneously optimise objective function and constraints, which also achieved a global performance index of the problem.

3.3.4 Goal Programming Method

The goal programming method is one of the fundamental methods of solving multi-objective optimisation problems. Many authors have given excellent definitions about the method with respect to its usefulness in solving problems. To this effect, the author will refer to a definition made by Saber & Ravindran, (1996), that "the method is used for solving problems with conflicting objectives, in which the user provides a goal or targets of achievement for each objective and prioritizes the order in which the goals have to be achieved. It then finds an optimal solution that satisfies as many of the goals as in the specified order". The mathematics surrounding this method is detailed in Mohammad, (2013). Such mathematics has limitation to particularly address linear functions. However, it can still be applied to solve specific industrial problems. This limitation cannot be said to have an adverse effect concerning this method hence, the linear programming technique can be used to solve large nonlinear optimisation problems (Ravi, 2005). Additionally, the goal programming problems are mostly present in design due to the number of constraints. Some of these constraints if not all, maybe in mutual conflict where by finding for example, a vector Z becomes impossible where all the constraints are satisfied. The objective of the above problem might be to locate the solution that approaches the target values of constraints or goal as closely as possible.

Assuming d_k^- and d_k^+ represent the deviations from the constraint or functional requirement target value of b_k, the objective is therefore to minimise;

$$U = \sum_{k=1}^{k} \rho k \, (w_k^- d_k^- + w_k^+ d_k^+) \qquad (3.4)$$

Where:

U is the objective function, ρk is priority for goal constraint k; $d_k^- \; and \; d_k^+$ are deviations from the target value of b_k; $w_k^- \; and \; w_k^+$ are weights associated with deviational variables $d_k^- \; and \; d_k^+$.

In this problem (goal) formulation, each goal objective is now represented below;

$$c_k(Z) + d_k^- - d_k^+ = b_k \qquad (3.5a)$$

$$0 \leq d_k^-, \; d_k^+ \leq 1 \qquad (3.5b)$$

$$d_k^- \times d_k^+ = 0 \qquad (3.5c)$$

$c_k(Z)$ is the goal expression, and it is normalised by division with appropriate value. $d_k^- \; and \; d_k^+$ are the deviational variables. Only non-negative values can be used for these deviational variables for the expression above to be meaningful, as these variables represent the achievement of the target value b_k . Achievement of goal is in two ways; that is under and over-achievement, but cannot possibly achieve both goals simultaneously. In this case, either one or both of the deviations must be a zero value. Thus, the weighted sum of deviations from goals is minimised. For goals that can be exceeded, that is $c_k(Z) \geq b_k, w_k^+$ would be zero while w_k^- would be 1, since it is needless to minimise over-achievement, but needful to minimise under-achievement. Given the same similitude for thresholds that are to be kept within the associated weights, $w_k^- \; and \; w_k^+$ will be 1 and zero. That is, w_k^+ would be 1 and w_k^- would be zero. Therefore, in accordance with the above explored dynamics, experimenting the choice of

priorities and weights is necessary in order to obtain the solution that reflects the preferred value system, because they dictate the solution.

Furthermore, the appropriate values of priorities and weights to assign to the deviational variables would solely depend on what is expected of a design. For instance, if the aim is to arrive at a design with certain key performance characteristics, then the rational procedure would be to assign higher relative priorities to appropriate deviational variables corresponding to those requirements. Thus, the approach is in contrast with the traditional optimisation approach to problems, where the values of the independent design variables dictate the solution. In this method, it is the values of the deviational variables that determines the values of the independent design variables.

Based on Archimedean (i.e., regarding parameter estimation) formulation Hering & Ulrich, (2012), the priority values may be one of two types or a combination of them:

$$\sum_{k=1}^{k}(w_k^- + w_k^+) = 1 \; and \;\; \rho k = 1 \; for \; k = 1,2,\dots,K \tag{3.6}$$

Again, ρk is chosen such that they represent pre-emptive priorities, i.e.

$$\rho 1 \gg \;\; \rho 2 \dots \gg \rho K \; and \; w_k^- = w_k^+ = 1.0 \; for \; k = \; 1,2,\dots,K \tag{3.7}$$

3.3.4.1 Generic Goal Programming Problem

The general goal programming mathematics of a problem is shown below. In its nature, it can obviously require satisfaction of some constraints to ensure feasibility as formulated:-

Find

$$\{x_1, x_{2,} \dots, x_N\}^T, d_k^-, d_k^+, k = 1,2,\dots,K \tag{3.8a}$$

(Decision and deviational variables)

Subject to:

$$g_i(Z) \geq 0, i = 1,2,\dots,I \tag{3.8b}$$

(Both linear and nonlinear inequality constraints)

$$h_i(Z) = 0, j = 1,2, \dots, J \qquad (3.8c)$$

(Both linear and nonlinear equality constraints)

$$c_k(Z) + d_k^- - d_k^+ = 0, k = 1,2, \dots, K \qquad (3.8d)$$

(Both linear and nonlinear goal constraints)

$$Z_{min} \leq Z \leq Z_{max}, 0 \leq d_k^-, d_k^+ \leq 1, k = 1,2, \dots, K \qquad (3.8e)$$

(bound on both design and deviational variables)

Minimise

$$U = \sum_{k=1}^{k} \rho k \left(w_k^- d_k^- + w_k^+ d_k^+\right) \qquad (3.8f)$$

(Weighted sum of deviational variables)

Priority type of the formulated problem can go in two ways: Pre-emptive or Archimedean as earlier stated. One of the successful applications of this method in problem solving was likened to Bertolini & Maurizio, (2006) who defined the best strategies for the maintenance of critical centrifugal pumps in an oil refinery using this method.

3.3.5 Trade Off On Pareto Front Method

This methodology is more recent, no weight system is used and the search process forms the Pareto frontier, or its approximation. It consists of splitting the solution procedure into two phases. The first phase selects the set of non-dominated trade-off solutions (Pareto frontier solutions) within the whole space of feasible ones with respect to the constraints. As stated earlier, a solution is considered as non-dominated if it is better than the others with relation to at least one objective. The second phase handles solutions belonging to the Pareto optimal, evaluates and compares them for the purpose of selecting the best. That is, the decision maker evaluates the generated design alternatives against the assessment criteria and looks for trade-

off solutions, subject to high computational time and effort. The reality of this method was demonstrated by Chapman, (2014), who explored Pareto front method and allowed inferior solutions to be removed from further consideration.

3.3.6 Convex Optimisation Method

The convex optimisation method is relatively one of the newest methods, with a well-defined mathematical optimisation approach towards getting a reliable and efficient solution. A lot of problems can be solved using this method provided the problems can be formulated as a convex optimisation. Whilst *Maxim*, (2014) formulated the convex problem and added that it can be applied to a wide variety of cost functions beyond classical formulations, Michael & Jon, (2014) demonstrated the usefulness of the convex optimisation framework in solving many different cost function problems. Several authors including Mohammed *et al*., (2014) have also done work using this method. Other methods of optimisation include but is not limited to linear least square method, genetic algorithm (*GA*) and the upgraded version of genetic algorithm methods otherwise referred to as Non-dominated sorted genetic algorithm (*NSGA-II*). An extensive literature on how *GA* and *NSGA-II* can be implemented for optimal functionality is explored in the later section of this chapter.

3.4 Techniques of Solving Multi-Objective Optimisation

There are different types of techniques such as Tabu search, Vector Evaluated Genetic Algorithm (*VEGA*), Multi-Objective Genetic Algorithm (*MOGA*), Niched Pareto Genetic Algorithm (*NPGA*), Weight-Based Genetic Algorithm (*WBGA*), Random Weighted Genetic Algorithm (*RWGA*), Strength Pareto Evolutionary Algorithm (*SPEA*), Improved Strength Pareto Evolutionary Algorithm (*SPEA 2*), Pareto-Archived Evolution Strategy (*PAES*), Pareto Envelop-based Selection Algorithm (*PESA*), Region-based Selection in Evolutionary Multi-objective Optimisation (*PESA-II*), Multi-objective Evolutionary Algorithm (*MEA*), Micro-GA,

Rank Density Based Genetic Algorithm (*RDGA*), and Dynamic Multi-objective Evolutionary Algorithm (*DMOEA*).

Other techniques, however, have been used in getting optimal solutions of a multi-objective problem. There are credible algorithms that have been used in a variety of applications, and their performances have been promising in several comparative studies. Whilst one of such studies can be found in Konak, (2006), who extensively discussed most of these techniques featuring their merits and demerits, the overview of a few of these techniques are explained below with relevant references from the literature indicating their areas of application.

Tabu search (*TS*) technique is a meta-heuristic search proposed by Glover in the 1980's for the purpose of solving combinatorial optimisation problems. It is ordinarily driven by simple but very efficient ideas to get almost optimal solutions for various types of difficult combinatorial optimisation problems. The technique is dominated by neighbourhood solutions in searching for optimal solution. Unlike *GA*, *TS* is highly dependent on the values of the algorithm's control parameters (Liang & Chao, 2008). The search commences when the parameters are chosen and a feasible solution to the problem is generated. Accordingly, the search operator referred to as 'move' can be altered in order to generate neighbourhood solutions. This operator has the tendency to place each element to move from its ideal location in the solution to any other alternative location within the solution space. Therefore, a set of neighbouring solutions are generated from 'move' through a pre-defined change to the current solution, in order to pave way for the best solution to be selected. The best solution is then selected from the current set of neighbouring solutions and becomes the new current solution. In this order, a new set of neighbouring solutions is further generated specifically from the new current solution. The process is recursively applied until the stopping criteria are achieved. Tabu list restrictions and the aspiration criteria of the solution associated with these restrictions are two main components of TS (Katsigiannis & Geogilakis, 2008). With respect to Tabu list restriction, it

is referred to as the adaptive memory in the sense that, some attributes are temporarily fixed, provided they are in the tabu list. In addition, proper choice of the tabu list size is very critical to the algorithm's success, and it depends on the nature of the problem. Tabu lists are managed by recording 'moves' in order in which they are made. When a new attribute finds its way into the tabu list, the oldest one is then released. Aspiration criteria have the tendency of overriding tabu restrictions if a certain move is unacceptable. Also, when satisfied, the aspiration criteria can reactivate this move. It is therefore very important to appropriately make use of such criteria to enable the *TS* technique to achieve best performance levels. Several successful research works have been carried out in *TS* techniques subject area. Amongst them are; optimisation of production cost while minimizing the amount of power loss in the power system Naama *et al*., (2013), and optimal load distribution strategy problem for a cooling system constituted by multiple chiller water units (Zhang & Zhang, 2010).

Vector evaluated genetic algorithm (*Vega*) is a special multi-objective technique due to its straightforwardness. It was originally the idea of Schaffer in the mid 1980's to use *Vega* in solving problems in a manner that the selection process is carried out independently for each criterion. This includes performing mating and crossover across subpopulation boundaries. It is less time complex because the approach does not need to transform multi-objective values into one value, and does not equally calculate each individual's dominant level based on a dominated relationship (Zhang &Fujimura, 2010). In other words, it rather divides the population into sub-populations, each of which evolve towards a single objective. This makes *VEGA* straightforward and somewhat amazing at first glance. However, the characteristics of *VEGA* cause selection bias, therefore the quality of solutions obtained using this approach may not be fantastic hence, *VEGA* approach lacks diversity as a major drawback. *VEGA* was proved inefficient when compared with other methods whilst solving multi-objective process planning and schedule (*PPS*) problems, to determine a schedule solution (Zhang &Fujimura, 2010).

3.5 Non-dominated Sorting Genetic Algorithm (NSGA-II): A Method for Solving Multi-objective problems

As stated earlier *NSGA-II* is upgraded version of genetic algorithm (*GA*) method and therefore starts in similar approach as the *GA*. The *GA* methodology starts by generating a random population of *N* chromosomes (i.e., suitable solutions for the problem).

3.5.1 Genetic Algorithm (GA)

Recall that *GA* is a meta-heuristic search technique for solving optimisation (both single and multi-objectives) problems (Jones *et al*., 2002). The technique is inspired by evolutionary theory. In nature, unfit species (chromosomes) within their environment (population) are faced with extinction by natural selection. Stronger ones have better chances of passing their genes to future generations through reproduction operators. Subsequently, species carrying the correct combination in their genes becomes dominant. Random changes (mutation) take place sometimes during the evolution process, and the new offspring may be retained or got rid of depending on their fitness value (Okafor & Sun, 2012).

A *GA* commences with a series of initial solutions, referred to as initial population. This population is normally generated randomly. *GA* handles the individuals contained in the generation as parents for the basis of producing children to form a new generation. Each individual in the population is evaluated and is assigned fitness. The quality of population (the average fitness and the maximum fitness) increases generation by generation. Making children is by mating parents which is referred to as crossover (one of the reproduction operators). Individuals with good fitness in *GA* have more chance of being selected as parents, in order to increase the probability of producing children with good fitness. It is always very possible to reproduce a child that is fitter than its parents if the parents are not already the fittest choice

(global optima). Mutation is another *GA* reproduction operator that is based on a random selection and random change, targeting to explore all search domains (i.e., global search).

In GA terminology, a solution vector $x \in X$ is called an individual or a chromosome. Chromosomes are made of discrete units called genes. Each gene controls one or more features of the chromosomes. Originally, GA genes are assumed to be binary digits, but in the later implementations, more varied gene types have been introduced (Konak *et al*., 2006). Normally, a chromosome corresponds to a unique solution x in the solution space. This requires a mapping mechanism between the solution space and the chromosomes. The mapping is referred to as encoding. *GA* work on the encoding of a problem, rather than on the problem itself. In general, *GA* is a population- based technique that is capable of solving multi-objective optimisation problems. Several works have been done using *GA*, including Moura *et al*., (2015), who used the *GA* technique to define an efficient inspection program in terms of inspection cost and risk level that complied with restrictions imposed by international standards.

The procedure of the *GA* methodology according to Sadrzadeh, (2012) is given as follows:

First step: Set $t = 1$. Randomly generate N solutions to form the initial population, P_1. Evaluate P_1 solutions fitness.

Second step: crossover. Crossover operation is simply to make children by mating parents. In general, the operation takes place on two parent chromosomes at a time and it is also referred to as binary variation. It generates offspring by combining the features of both parents through exchange of genes. Since it is not unusual for child chromosomes to inherit some features of the parent chromosomes, the crossover operation is applied in anticipation of producing better sets of the existing parent with stronger fitness, thus this is dependent on the environment. Take for example, a family where one parent has exceptional liking for a Chemistry subject, and the other parent has excellent English communication skills. When the chromosomes of these

parents are combined, the progeny will be produced comprising of the both parents features, where it will be expected that at least one of the children will have both a liking for chemistry, and excellent English communication skills. So if the fitness is anchored or judged on the grounds of these acquired skills (environment), then the new child will be valued over either of its parents. The inheritance of good genes from parents to child chromosomes plays an effective exploration role in the search space. This is because each new and fitter child chromosome is in fact, a new point (solution) in the total search space.

There are different types of crossover operations, thus depending on the type of chromosome representation. For array or typical string chromosomes representation, Single-point, multi-point and uniform types of crossover are mostly adopted. These crossover types usually have defined crossover probability, as successful crossover is the one with defined crossover probability. However, probability of crossover is highly problem dependant so the parameter deserves to be selected properly to suit the problem. Improper selection of this parameter can result in the following;

- Premature convergence (that is, the solution is trapped in a local optima) and
- Delay in solution convergence (that is, an inefficient algorithm).

Solution trapped in local optima is not in the best interest of the *GA* objective, particularly as regards to optimisation. The overall purpose of an optimisation is to find the best value of a function after taking into account all relevant parameters and constraints. This value literally dominates all other possible values of the function in the solution space, thus it depends on the optimisation objectives. For an optimisation objective that has to do with minimisation, then the best value gotten from the optimisation is exceeded by all other values of the function. Similar to this, if the optimisation goal is based on maximising the objective function, the best derived value arising from the optimisation process exceeds all other possible values which may exist on the solution space of the function. An Optimisation process with appropriately

defined and applied crossover parameters that represents such desirous characteristics of either being minimised or maximised is regarded as the 'global optimum' of a function. In a nutshell, whilst a global optimum represents a point in a search space where all other points are either worse or equal to the best value, local optimum represent the best solution for each region. The importance of this is occasioned in most situations where the search space is wide and divided into various regions; the global optimum for a particular region may not be uniform with regards to the global optimum of the other neighbouring regions. For this reason, the candidate further refers reader to Figure 3-2 where clear demonstration concerning global optimum and local optimum solutions are extensively demonstrated.

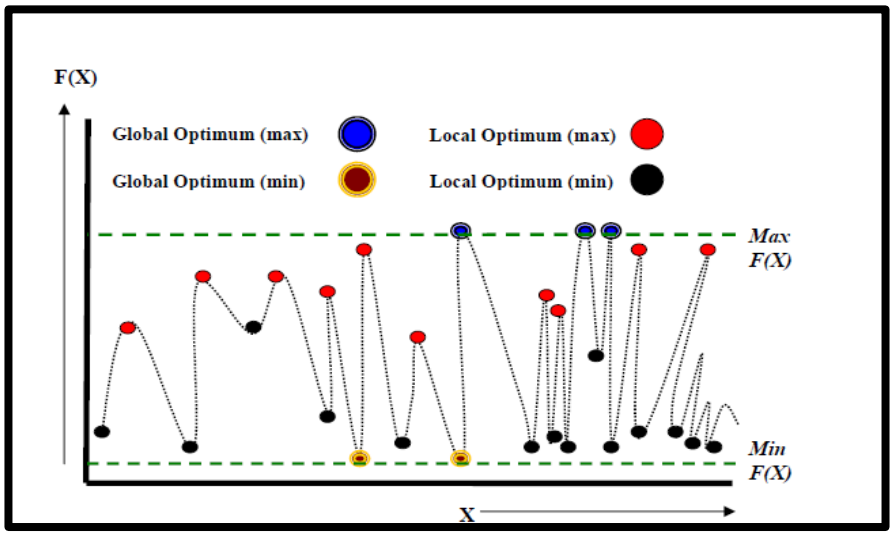


Figure 3-2 A Function *F(X)*, global optimum and local optimum

Figure 3-2 gives credence to the earlier explanation as it shows different types of function's optimum values. It shows that global optimum can be maximum and minimum, as well as local optimum. For Maximum function $F(X)$, the global optimum is located in three distinctive regions within the design search space, which also shows the different values of X variables. And for minimisation of function $F(X)$, the Figure clearly demonstrates two instances at which the referred global optimum values can be found. In general, the second step generates an offspring population U_t as follows:

- Choose two solutions x and y from P_t based on the fitness values.

- Using a crossover operator, generate offspring and add them to U_t.

Third step: Mutation: Mutate each solution $x \in U_t$ with a predefined mutation rate.

Fourth step: Assignment of fitness: Evaluate and assign a fitness value to each solution $x \in U_t$ based on its objective function value.

Fifth step: Selection: Select N solutions from U_t based on their fitness and copy them to P_{t+1}.

Sixth step: If the stopping criterion is satisfied, terminate the search and return to the current population, otherwise, set $t = t + 1$ go to second step.

With respect to the GA Selection process, there are different types of selection processes. Thus, the application of them depends on the nature of the problem. Amongst them are; Random selection of generated solutions, direct selection of first generated fittest solutions, Deterministic selection, Tournament selection, Elitism Selection and Roulette Wheel selection process. Parents can successfully be selected in readiness for crossover. During selection, it is however suitable to have chromosomes with the potential of providing best solutions given the most opportunity, while the less suitable chromosomes should not be completely eradicated from the population in order to ensure diversity in the population (Xue & Wang, 2015). *GA* can also be somewhat challenging in terms of having better solutions due to early solution convergence, or trapped in local optimal if inappropriate reproduction operators values such as probability of crossover and probability of mutation are selected. In most possible cases, *GA* may even have difficulty in handling constraints.

However, given the exceptional flexibility of *GA* as explored in the literature, *GA* has:

- Flexibility in modelling of engineering problems. That is, it has no strict mathematical requirements, such as derivative requirement, on the objective functions and constraints. The only requirement is that the objective function constraints can be evaluated in some

way. Additionally, *GA* is also suitable for dealing with those problems including discrete design variables.

- Global optimisation ability, *GA* has been recognised as one of the most effective approaches in searching for the global optimal solution (Zhigang & Ming Zuo, 2006).

3.5.2 Characteristics of NSGA-II in Handling Multi-Objective Problems

NSGA-II handles multi-objective optimisation algorithms with three special characteristics; fast non-dominated sorting approach, fast crowded distance estimation procedure and simple crowded comparison operator. In addition, it provides a much better spread of solutions and better convergence near Pareto-Optimal front (Kalyanmoy *et al*., 2002). NSGA-II algorithm adopts the fast non-dominated sorting technique and a crowding distance to rank and select the population fronts. In order to ensure elitism, *NSGA-II* employs the standard bimodal crossover and polynomial operators in combining the current population together with the generated offspring as the next generation. The best individuals in terms of non-dominance and diversity are then selected as the solutions. For example, a front for a population member (a solution k) is determined by the number of solutions that dominate k and the number of solutions dominated by k. The fitness of an individual is assigned according to its front. Based on that, the population is then filled up with solutions from the remaining fronts when the size of the first front is smaller than the size of the population. Clearly, a few fundamental advantages about *NSGA-II* are; (a) it avoids the difficulty of setting sharing parameters, and equally has a low time complexity of $O(NlogN)$, where N is the population size (Huang *et al*., 2010). $O(NlogN)$, means that it can carry out sorting very easily. An orderly explained *NSGA-II* algorithm flow chart is shown in Figure 3-3.

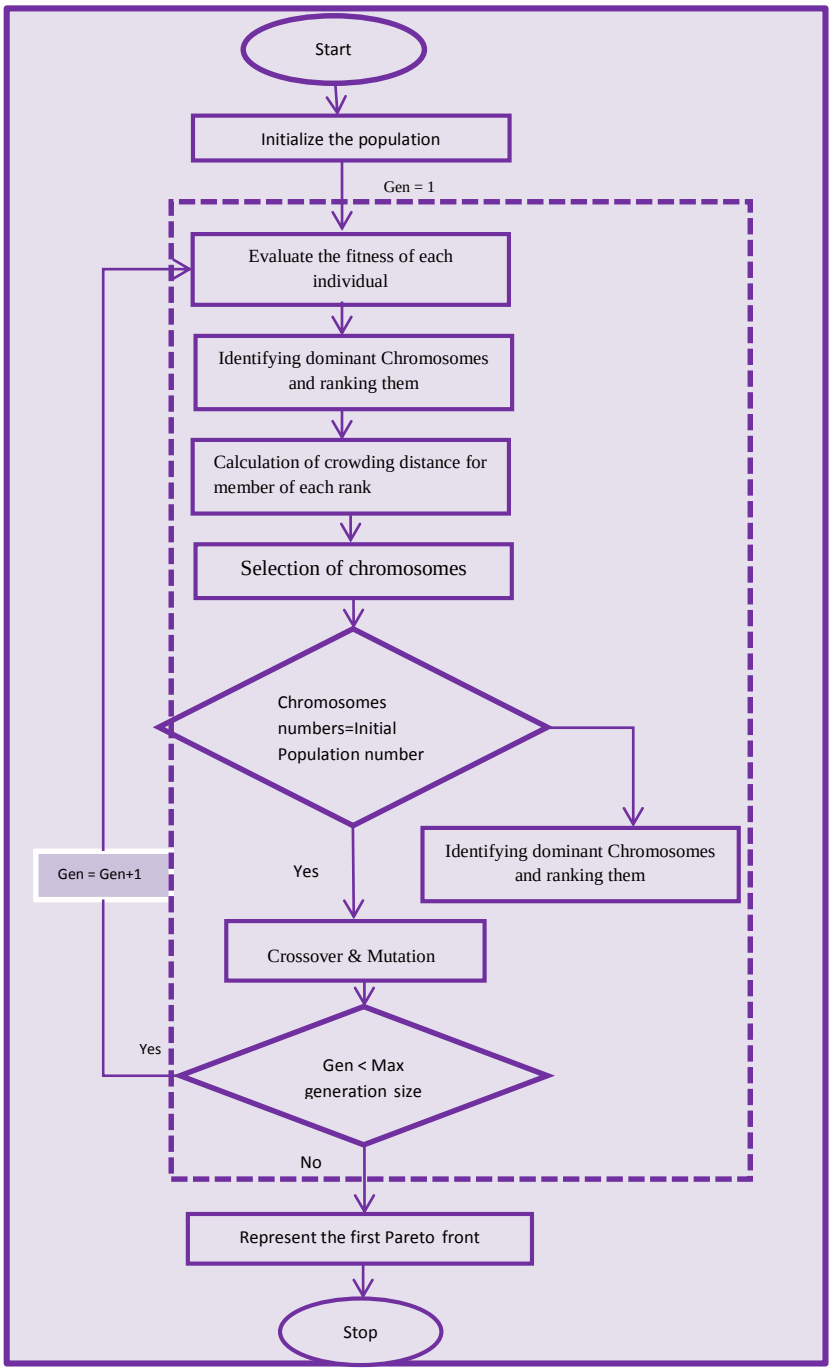


Figure 3-3 NSGA-II Flowchart (Zahra *et al.*, 2015)

3.5.3 Fitness:

Based on Figure 3-3, having generated the initial population as stated earlier, it is normally followed with fitness evaluation of each multi-objective chromosome in the population. This is mainly to ascertain how close a given design solution is able to achieve the set aim. As stated above, each designed solution is commonly represented as a string of numbers (chromosome).

Fitness idea is to get rid of the worst design solutions, whilst the best design solutions are accommodated in the new generation. Based on this, each solution is awarded a figure of merit (i.e., a quality used in characterizing the designed system performance relative to its alternatives). Very often in *NSGA-II*, the figure of merit adopted is with respect to maximum and minimum functions (i.e., *Maximin*) as first proposed by Richard Balling (Balling, 2000 & Balling, 2003).

A fitness function problem is formulated below for a given solution. Considering a minimization problem with k objectives to solve based on a population based algorithm, the fitness of i^{th} individuals in the population with *Maximin* fitness function approach becomes:

$$fitness_{\ i} = max_{j \neq i}\left(min_k\left(f_i^i - f_k^j\right)\right) \tag{3.9}$$

The *Min* function is gotten from all the design objectives, while the Max function is gotten from all the individuals in the population (Alemzadeh & Dastghaibyfard, 2013).

In addition, the *Maximin* fitness function enhances in getting the following information:

- The information that identifies exact dominated individuals ($fitness_{\ i} > 0$) as well as non-dominated individuals ($fitness_{\ i} < 0$) . In other words, the weakly dominated($fitness_{\ i} = 0$).
- Rewards non-dominated solutions in less crowded regions and penalizes the crowded individual found in the crowded areas. That is; to carry out clustering of non-dominated solutions.

However, the shortcomings with *Maximin* fitness function is the inability to identify better solutions to keep for the next generation in a situation where a multiple non-dominated solution is achieved under the same *Maximin* fitness function. Additionally, the fitness of a non-dominated solution may be adversely affected by a dominated solution if the fitness function

is not modified. In the year 2001, Balling further modified the fitness function to avoid adverse effects on solutions by ensuring that *Maximin* fitness function of a non-dominated solution is unable to be controlled by non-dominated solutions. The modified *Maximin* fitness version is as follows:

$$fitness_{\ i} = max_{j \neq i, j \in N} \left(min_k \left(f_i^i - f_k^j \right) \right) \tag{3.10}$$

where N is the size of Non-dominated solutions.

3.5.4 Ranking

In conventional multi-objective problems, where a non-dominated solution is achieved, each chromosome is assigned a rank, and it is important to maintain a good spread of the solutions in the optimum fronts. For this purpose, *NSGA-II* uses the crowed-comparison mechanism for preserving diversity among the population members. And for every chromosome in a Pareto front, a crowding distance is ascertained as the distance of the largest cuboid contacting the two neighbouring solutions, as depicted in Figure 3-4 (Ali & Arezoo, 2014).

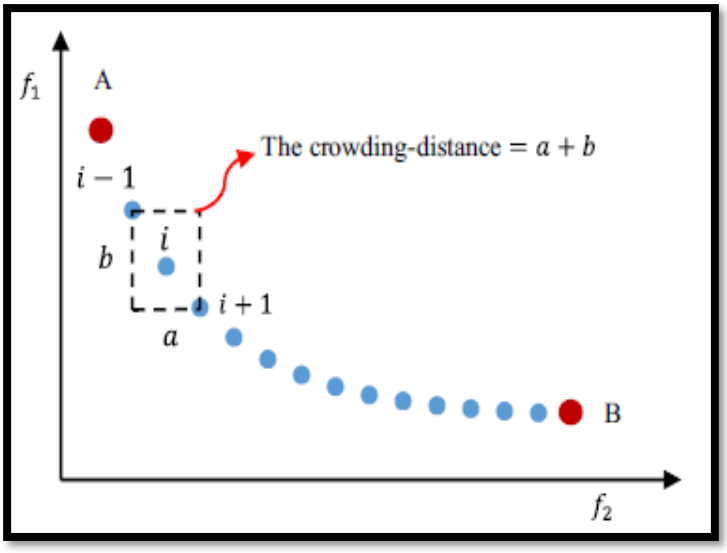


Figure 3-4 NSGA-II crowded distance (Ali & Arezoo, 2014)

The above Figure 3-4 shows that, for single objective function, the boundary solutions with the maximum function value point A and a minimum function value point B are usually given large distance values that may term to infinity, and these solutions, which are in the extremes of the non-dominating front, deserve to be emphasized more than the immediate solutions. To emphasize that, the distance space between two consecutive points in the Pareto set must be measured using Equation (3.11)

$$\mathrm{d}_i^{sp} = \sqrt{\sum_{k=i}^{m} (f_k\ (\mathrm{x})^{max} - f_k(\mathrm{x})^{min})^2} \qquad (3.11)$$

where;

d_i^{sp} is the distance between successive Pareto fronts.

m is the maximum number of objective functions.

$f_k\ (\mathrm{x})^{max}$ is the maximum of the k^{th} objective between the two consecutive points in the Pareto set.

$f_k(\mathrm{x})^{min}$ is the minimum of the k^{th} objective between the two consecutive points in the Pareto set.

Also, to account for the extremities of the solutions in a standard *NSGA-II*, a new matric capable of measuring every distance between all solutions and the extreme points in the Pareto set is defined. This defined metric is referred to as closeness-distance, see Figure 3-5.

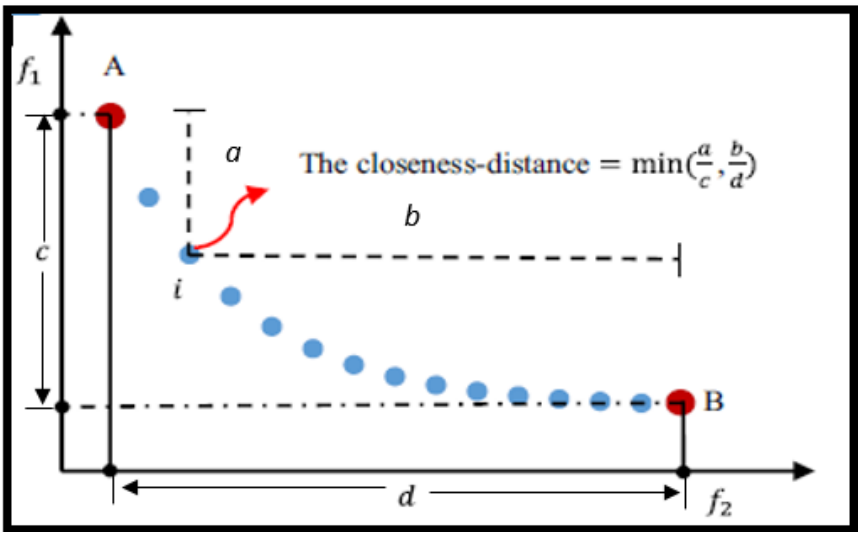


Figure 3-5 *NSGA-II* Closeness-distance (Ali & Arezoo, 2014)

Figure 3-5 depicts *NSGA-II* closeness-distance, whilst Equation 3.11 is the computation of the closeness -distance which considers every individual i $(i = 1to\ N)$. With respect to every objectives function $(j = 1\ to\ k)$, it is given by Equation 3.12 in order to be ranked.

$$\mathrm{d}_{i,j} = \frac{f_j^i - f_j^{min}}{f_j^{max} - f_j^{min}} \tag{3.12}$$

From Equation 3.12, $\mathrm{d}_{i,j}$ is the normalised dimensionless value that differs from Zero in the boundary solution A to 1 in the most distant solution. In this case, the most distant solution from Figure 3-5 is solution B in the Pareto front. While f_j^{min} and f_j^{max} are the minimum and the maximum values of the objective function j respectively in the population, and f_j^i is the value of the objective function j in the solution i. The boundary solutions between two extreme solutions, as well as the solutions close to them, are further emphasized through a selection process, which only takes place after new population has been created.

3.6 Chapter Summary

Summarily, traditional redundancy allocation problem (*RAP*) is simply a multi-objective optimisation problem (*MOOP*). Therefore, addressing *RAP* using any suitable *MOOP* approach is a welcome idea. In addition to that, the candidate presented a variety of approaches and techniques of *MOOP* and established that the *NSGA-II* method was the most suitable to adopt for successful delivery of this project hence, the approach is flexible and gives a better spread of solutions, gives a good level of convergence if selection criteria are handled especially in terms of assigning *pc* and *pm* values. Assignment of improper *pc* and *pm* can result in premature convergence or solution been trapped in a local optimum. Therefore, given these highlighted features of *NSGA-II*, in the following chapter four, *NSGA-II* methodology is adopted for multi-level formulation. This formulation constitutes the delivery of this project aim, and it has been proven as a general redundancy allocation problem.

4. Multi-Level Formulation

4.1 Introduction

Multi-level formulation (*MLF*) is a process of formulating a system with its associated components to various levels. It is a redundancy allocation problem (*RAP*), which can be difficult to solve. *RAP* is a nondeterministic polynomial-time hard (NP-hard) problem, meaning it is difficult to solve. To solve *RAP*, many proposed heuristic and meta-heuristic methods have only been successful on a single level system. For a multi-level system, as referred in this chapter as multi-level formulation (*MLF*), *RAP* challenges still persist. A high level reliability is certainly a crucial target in *MLF*. Reliability of components in such a system is usually fixed, and one of the alternative ways to enhance system reliability is simply to allocate more redundancies in parallel. To achieve that, usually normally attracts additional resources such as cost and materials.

4.2 Multi-Level Formulation State-of –the Art

The literature surrounding *MLF* is simply evolutionary. They evolve alongside with the technology. That is, as technology evolves across the board, it gives room for modern systems to consist of several functionalities and levels. This evolution also places more challenges on research. In recent times, research concerning *MLF* has now gradually extended to other domains beyond the study of combining a purely parallel or series connection to a larger scale component combination, under several limitations such as cost, weight, and volume. *MLF* is more complicated considering that replacing components in the sublevels can directly influence its upper and lower levels in certain cases. For example, series system case under usual stochastic order redundancy at component level can be better than redundancy at system levels

for series systems. However, this is not applicable on parallel systems as redundancies on parallel system level is as good as the redundancies at components level.

To date, under *MLF*, once a decision is reached on redundancy, it becomes necessary to examine how deployment of redundancy can successfully be made. Systems get examined, and depending on the examination outcome, a decision to either duplicate the entire system or only duplicate a certain component that is identified for duplication can be decided. If any of the components are identified for duplication, then the same component is duplicated in parallel to actualise system reliability. Moreover, reliability can be increased in a variety of ways, but the choice absolutely depends on the nature of the system. In certain systems, the use of redundancy may not provide the optimal solution, considering systems in which the minimum size, cost and weight are overriding considerations and objectives. A major challenge in redundancy allocation is therefore, how to handle the impacts (i.e., increase in cost and weight) created as a result of additional components on the system for reliability purposes. Whilst earlier works have been mainly limited to single objective optimisation of simple systems (Painton & Campbell, 1995), more recently challenges due to configuration of series, parallel and complex (Hsieh & Yeh, 2012) and Sun et al., (2008) and state of configuration "static/dynamic" (Ahmed & Abul, 2011) have been tackled.

4.2.1 Reliability Model for Multi-Level Formulation

A typical redundancy allocation model of a system involves multiple hierarchical levels. That is, from the topmost level, down to the lowest level. These levels are the topmost level, which represents the system. The module level is located between the topmost level and the second lowest levels, and finally the lowest level which represents the component level. Modules can have any number of subordinate components to form a system as depicted in Figure 4-1. The Figure depicts a typical general multi-level redundancy allocation showing the logical relationship that exists among the modules at different levels, in which some maybe in series,

parallel or a combination of both. This makes it necessary to always understand, for example which system configuration is to undergo reliability calculation. For a series- parallel system, the mathematical formulation is shown in Equation 4.1.

$$\left\{max\left[R=\prod_{i=1}^{s}\mathrm{R}_i\left(\mathrm{x}_i\right)\right], min\left[C=\sum_{i=1}^{s}\sum_{j=1}^{m_i}c_{ij}x_{ij}\right], min\left[W=\sum_{i=1}^{s}\sum_{j=1}^{m_i}w_{ij}x_{ij}\right]\right\} \qquad (4.1)$$

subject to $\quad 1 \le \sum_{j=1}^{m_i} x_{ij} \le n_{max,i} \quad \forall\, i = 1, 2, \dots, s$

$$x_{ij} \in \{0, 1, 2, \dots, s\}$$

where *R, C, and W* are the reliability, cost and weight of the system respectively, *s* is the number of modules, x_{ij} is quantity of j^{th}component in module i, $n_{\max,i}$ is user defined maximum number of components in parallel used in module i, m_i is total number of available components for module $i, R_i(\mathrm{x}_i)$ is reliability of module i, $while\ c_{ij}, w_{ij}$, are cost and weight for the j^{th} available component for module i respectively.

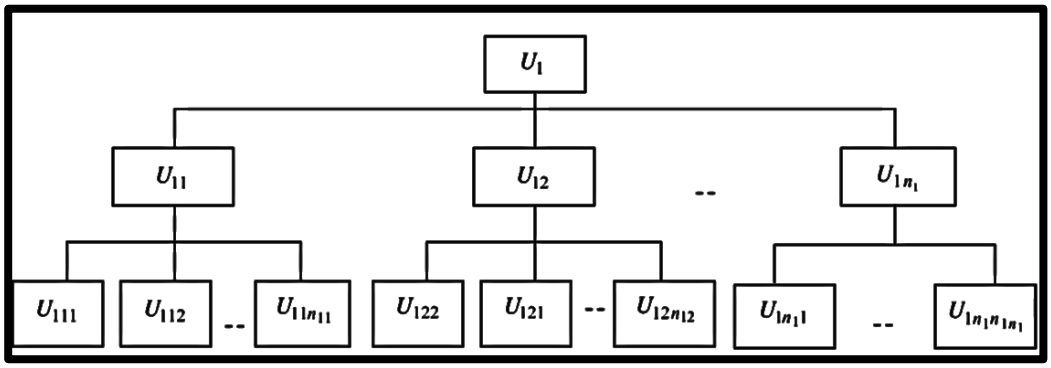


Figure 4-1 A general multilevel redundancy allocation configuration

U_1 is the system containing U_{11} to U_{1n_1} modules at its next lowest level. In a similar arrangement, the U_{11} module contains n_{11} sub-units as components at its next lowest level,

represented as U_{111} to $U_{11n_{11,}}$ which is truly the second hierarchical level of the system. The process can continuously be replicated logically until the actual lowest level of the system hierarchy is reached using connecting lines. The connecting lines logically indicate the relationship among the modules at different levels. It usually takes either series or parallel configurations or both, while all system levels provide reliability such that the reliability calculation takes into consideration the nature of the system configuration. In this case, to calculate reliability R_i of system U_1 for a multilevel series, system reliability can be achieved by;

$$R_i = \prod_{m=1}^{n_i}\left[1 - \prod_{j=1}^{x_i}\left(1 - R_{i,m}^{j}\right)\right] \tag{4.2}$$

Where, $R_{i,m}^{j}$ are reliability values for sub-units $U_{i,m}^{j}$, a unit in the $j-th$ redundant unit of the $m-th$ sub-sub unit of U_i. U_i is the $i-th$ unit (a common name for system, subsystem and component unit). x_i is the number of components used in U_i, and n_i represents the number of sub-units in U_i. Similarly, the reliability for a multi-level parallel system configuration can be calculated by:

$$R_i = 1 - \prod_{m=1}^{n_i}\left[\prod_{j=1}^{x_i}\left(1 - R_{i,m}^{j}\right)\right] \tag{4.3}$$

The parameter definitions of n_i, x_i, $R_{i,m}^{j}$, in Equation 4.3 remain the same as in Equation 4.2. For the very lowest level of a multi-level system, which is usually the least system level where no additional system level exists, the reliability can be obtained by:

$$R_i = 1 - \prod_{j=1}^{x_i} \left(1 - R_{i,}^{j}\right) \tag{4.4}$$

A system cost is as important as system the reliability. Usually, most systems with higher reliability are subject to cost, and a total system cost is best appreciated if it stays within the system pre-defined cost value. System cost under *RAP* for simple systems is usually the summation of each component's cost. For multi-level systems, additional costs are considered, and such cost consideration is normal as it reflects the existence of other levels (addition or duplication of redundant units) in the system. For this purpose, a pre-defined cost, alongside the total cost of system U_i in a multi-level that can account for additional material cost can be computed.

4.3 Multi-Level Formulation for a Series -Parallel System

4.3.1 Motivation

A series- parallel system reliability level is high and outperforms most other types of system configurations. For this reason, most industrial systems are currently designed in a series – parallel form to enjoy ideal system reliability in both component and system levels. Multi-level formulation, as previously discussed, is usually subject to optimisation constraints hence, it clearly handles *RAP* to actualise reliability improvement without limitations on any level. The motivation for this work is to do something reasonably different from the usual traditional method in this interest area by establishing a realistic integrated design methodology to address complex engineering problems. This is in contrast to traditional design methods concerning RAP, hence, all successful redundancy allocation work reported in the literature did not establish such integrated design methodology. One of the approaches to establish this method is through the use of *MATLAB* software tool which is widely used to solve most present

engineering problems. Details of *MATLAB* software tool functionalities with the user friendly manual is provided in Chapter five of this work, explaining what the program does, the tool and methodology used.

4.3.2 Basic Multi-Level Formulation of Redundant System Modules

A: Module.

<n, m> : Min/Max number of module components.

A1 <1, 3> : Module *A1* with min/max components.

: indicates the presence of system branch or sub-branch.

For example, considering that redundant module *A1* with defined minimum and maximum redundancy *A1<1, 3>* allocates its redundancies traditionally, the formulation will either be as follows: to have two components parallel to each other, and one independent series component. Alternatively, it can have all three maximum components in parallel, see Figure 4-2.

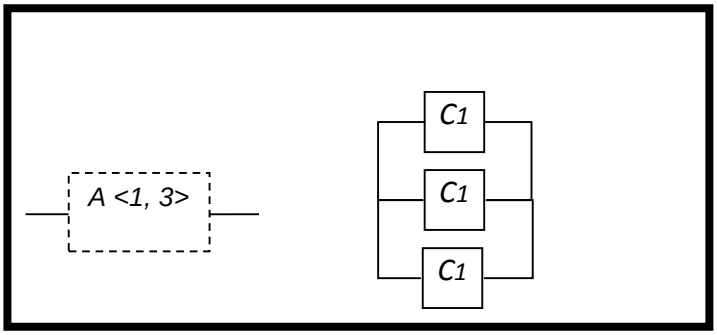


Figure 4-2 System module *A1* with three parallel components

Hence the components are allocated in parallel, the reliability Equation for Figure 4-2 is therefore shown in Equation 4.5 in page 110 of this work.

The system module will function provided one of the components (*C1*) is working. For example, assuming the reliability of each components (*C1*) is 0.95, the overall reliability of the module can be calculated using Equation 4.5.

There are however cases in which redundant modules (modules with components) may be associated with upper and lower levels as demonstrated in Figure 4-3, and in such a scenario the lower module can be treated as component.

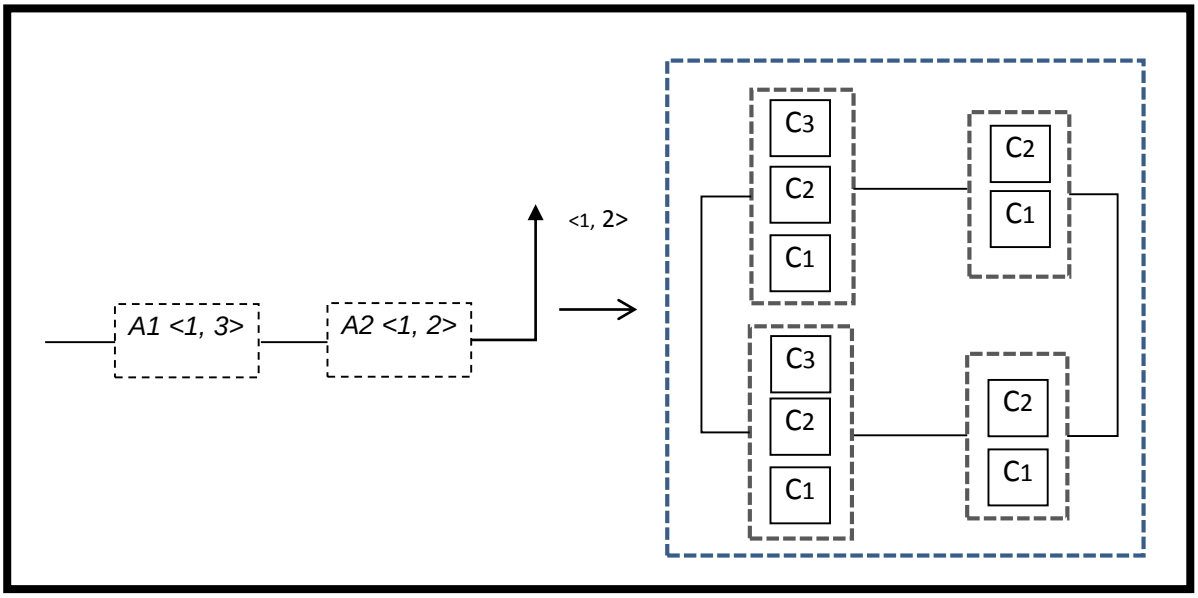


Figure 4-3 Components of modules *A1 - A2* with associated upper module

Figure 4-3 above shows a parallel allocation of components from modules *A1 -A2* with its associated upper modules. The reason for the upper module is due to the presence of modular branch associated in *A1-A2*. This branch is indicated with upward arrow, and it is further developed using blue rectangular dotted lines to show that module *A1-A2* with its associated upper modules belongs to a singular branch. The reliability Equation is the same as Equation 4.3 shown earlier hence the module components are allocated in parallel, and it is beyond one level. Therefore, in order to maintain orderly Equation numbering, in page 110 of this work, this same Equation is referred to as Equation 4.6.

4.3.2.1 Series – Parallel System: A Case Study for a Typical Multi-Level Formulation

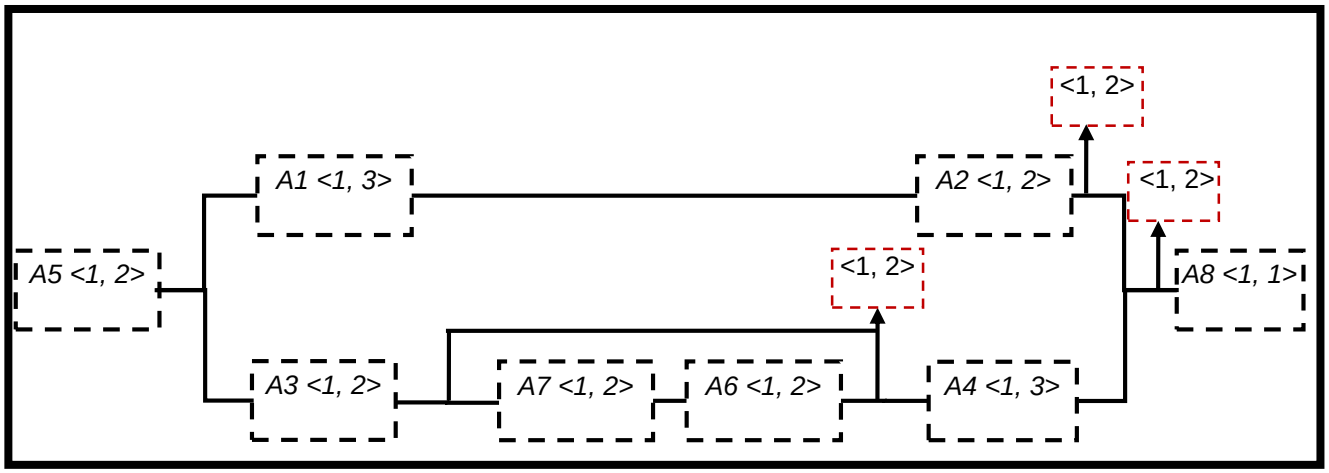


Figure 4-4 A series- parallel system.

The reliability of the system can be calculated using Equation 4.7 in page 110 of this work.

Figure 4-4 above is a series- parallel system developed using *MATLAB* configuration code as shown below:

The *MATLAB* configuration code of Figure 4-4**:**

'A5<1,2>+((A1<1,3>+A2<1,2>)<1,2>*(A3<1,2>+(A7<1,2>+A6<1,2>)<1,2>+A4<1,3>))<1,2>+A8<1,1>' **.**

The *MATLAB* configuration code of Figure 4-4 above shows a system with various modules, branches, sub-branches and a specific maximum number of components each module can have using *RBD* method. *RBD* is a good method for the modelling and analysis of a system.

From Figure 4-4, it shows the feasibility of allocating redundancy to multi-levels. The levels present on the system are:

- Redundancy at module level
- Redundancy at branch level without limitations on the number of levels
- Redundancy at sub-branch (part of system branch) level.

4.3.2.2 Analysis of the System Redundant Modules (*A1-A8*)

System modules usually consist of a set of components. In this case, modules *A1-A2* represent a full system branch, while modules *A7-A6* represent part of a branch (sub-branch) of the system. Generally, the system consists of eight modules *A1-A8* with redundancies. Each module selects components based on component availability and in accordance with configuration specifications. The configuration is made up of branches and a sub-branch (part of system branch), and each branch allocates redundancies in a similar way to that of modules (i.e., number of branches can be defined via components availability).

For example, modules *A2, A3, A5, A6 and A7* are redundant modules. However, each of these modules can only choose available components *(<n, m>)* between minimum of one and maximum of two <1, 2>, while redundant modules A1 and A4 are restricted to selecting components available between a minimum of one and maximum of three <1, 3>.

Redundant module A8 is only allowed to make component choice between a minimum of one and maximum of one <1, 1>. In addition, the branches and sub-branch present in the system can only select one component between a minimum of one and a maximum of two <1, 2>.

Whilst modules *A5* and *A8* are separately redundant for its module because the modules are part of the system, modules *A1-A2* are redundant for the upper system branch. Modules *A7-A6* are redundant for the sub-branch located at the system lower branch, and the modules *A3-A4* are redundant for the full lower system branch.

4.3.2.3 Multi-Level Formulation for Module, Branches and Part of a Branch (Sub-branch)

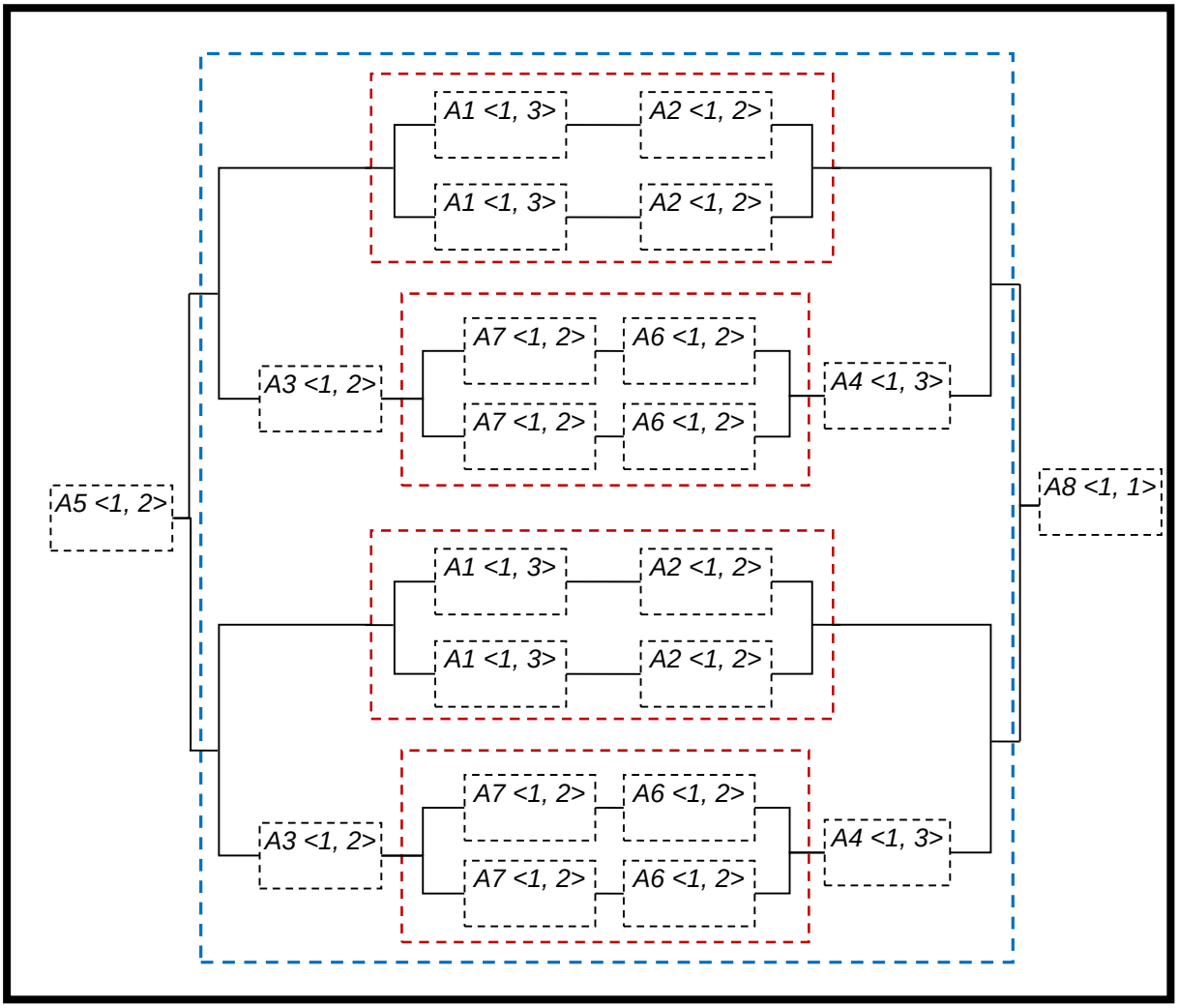


Figure 4-5 Defined Redundancies for modules, branches and part of branch (sub-branch)

Figure 4-5 shows redundancies at three levels (Modules, branches and sub-branch). Modules A1-A8 as indicated with black rectangular discrete dotted lines signifies modular redundancy. Redundancy at branches and part of branch (sub-branch) without limitation is signified with red and blue rectangular dotted lines.

The top most echelon of Figure 4-5, red dotted lines imply a full system branch with redundant modules *A1-A2*. This branch can only have redundant components chosen between a minimum of one and maximum of two <1, 2> as clearly indicated earlier in Figure 4-4.

Next to the top most branch, is redundant modules *A7-A6*. These modules are distinguished with another rectangular dotted red line, which in turn implies that modules *A7-A6* are a sub-branch (part of branch) to the systems lower branches. This distinction is necessitated to recognise Modules *A3-A4* as members of the same system lower branch but not belong to the sub-branch (part of branch).

A system branch precedes its sub-branches, and in most cases may have a discretely defined redundant component that supersedes a redundant component that is separately defined for its sub-branch without overlapping. Thus, in the above lower system branch of Figure 4-5 in discourse, there is no defined redundancy presence to base its component selection limits (i.e., minimum and maximum components in which the entire system lower branch should have). Rather, only its sub-branch has a defined redundancy of one as a minimum and two as a maximum <1, 2> as indicated in Figure 4-4 using the appropriate system/subsystem branch symbol (upward arrow) against modules *A7-A6*.

Subsequently, the larger rectangular blue dotted lines show that modules *A1-A2* and *A7-A6* which comprise branches and a sub-branch can only have the exact redundant components as defined in Figure 4-4. The defined redundancy is between one and two <1, 2>, and treated ideally as a branch based on the configuration. This was illustrated in Figure 4-5 above using rectangular blue dotted lines to indicate that it is a branch. And in Figure 4-6 overleaf, it shows that the branch has one as minimum, and two as maximum defined redundancies.

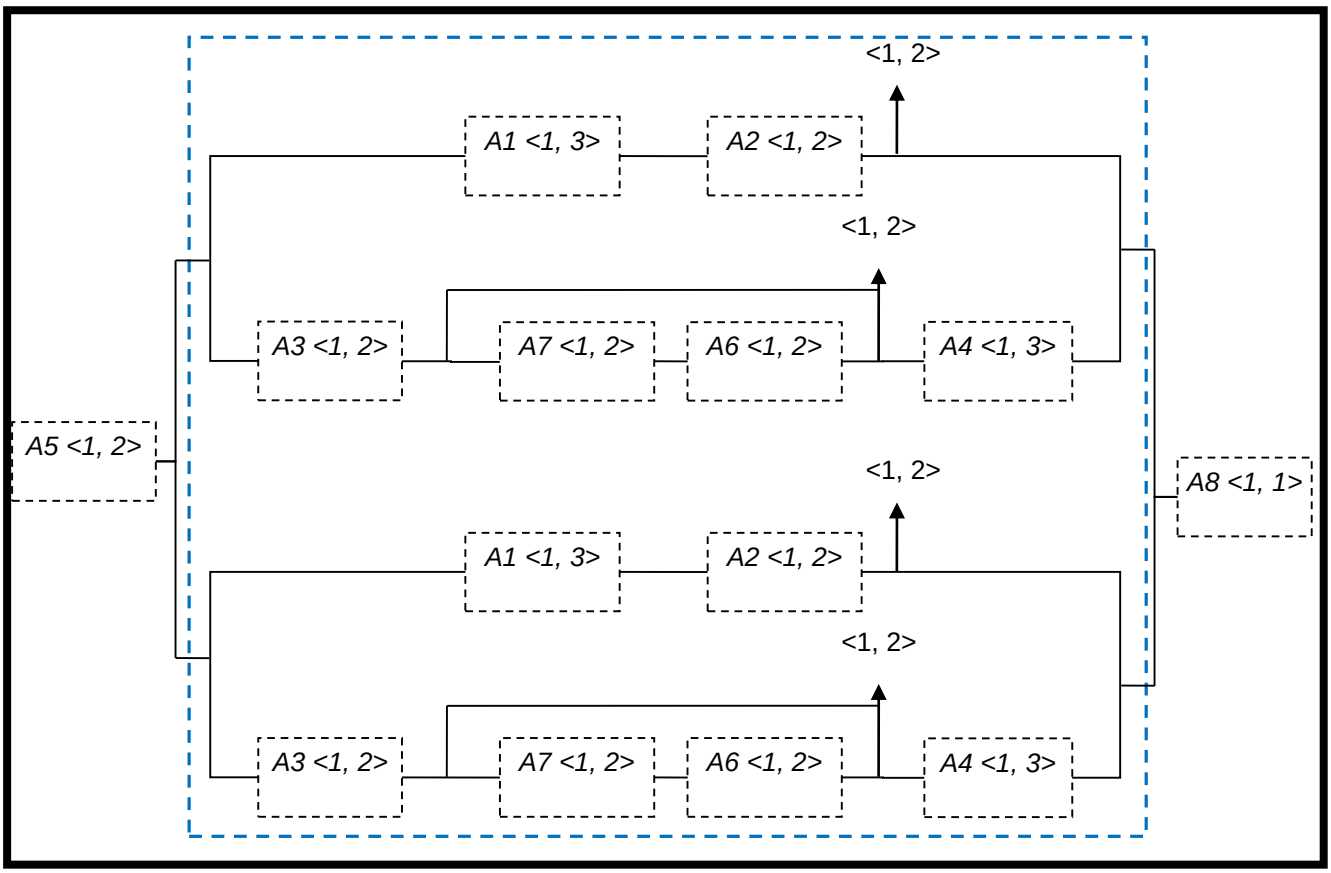


Figure 4-6 A Branch with maximum defined redundancy of two

Figure 4-6 shows that redundant modules *A1, A2, A3, A4, A6, and A7* have another branch level with defined redundancy of one and two <1, 2> as minimum and maximum components.

However, for clarity purposes, similarity exists between Figures 4-5 and 4-6 above. But the significant differences between the two Figures are as follows;

- Figure 4-5 detailed the system branches comprising (upper, lower and sub-branch) with defined redundancies for each, while
- Figure 4-6 acknowledged the existence of the various branch levels. It additionally demonstrated that they belong to a singular branch. The branch redundant components is between a minimum of one and a maximum of two <1, 2> as shown in the typical multi- level redundancy allocation in Figure 4-7.

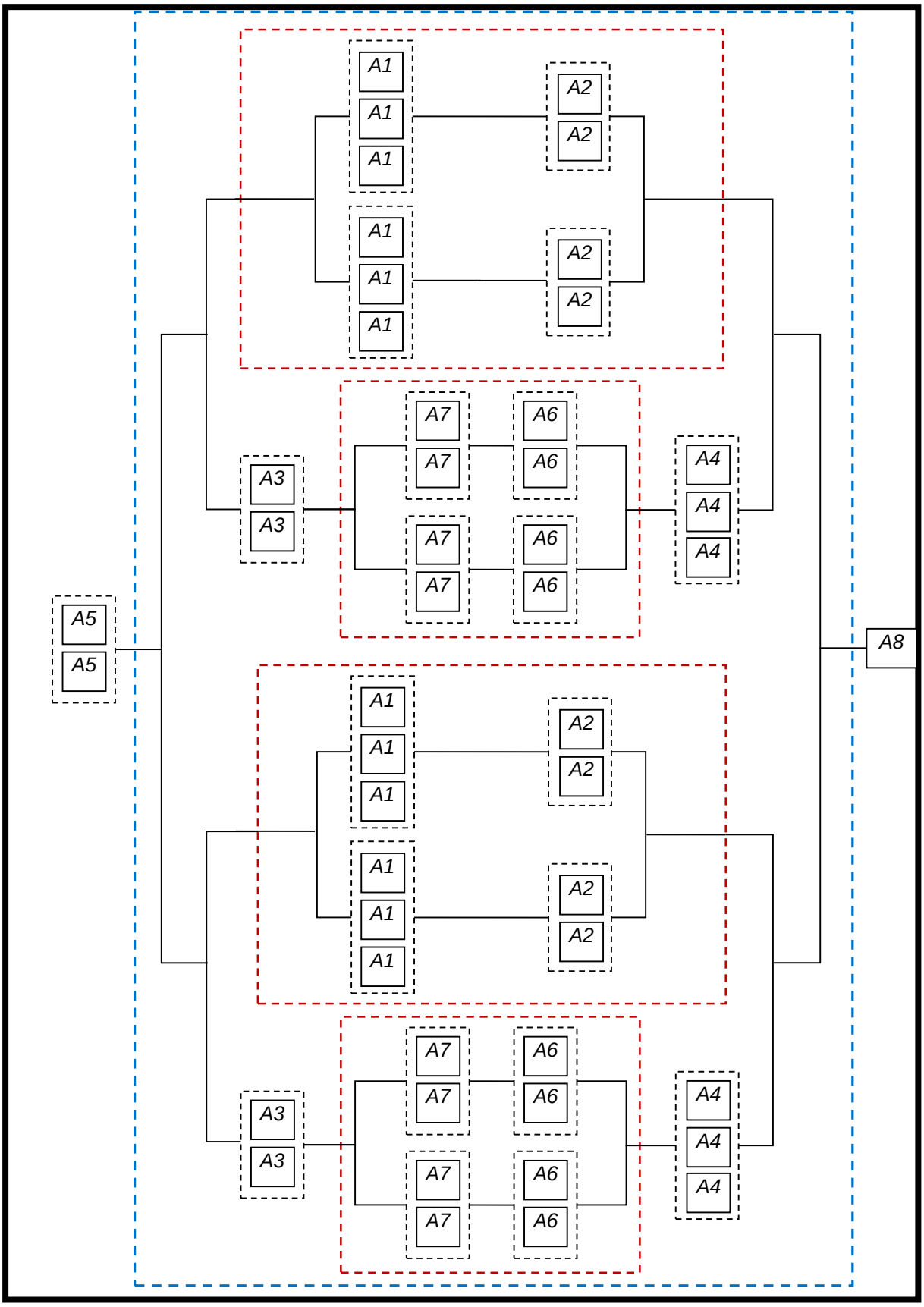


Figure 4-7 A Typical Multi level formulation

Figure 4-7 is a typical multi-level formulation of the configured system in Figure 4-4, and implemented in accordance with Figure 4-6 analysis, which has in overall a branch with maximum defined redundancies of two.

In addition to Figure 4-7, module *A1- A2* represents a full system branch with defined modular redundancies properly allocated. Whilst modules *A7-A6* a represents sub-branch (part of branch) with defined modular redundancies allocated properly, modules *A5* and *A8* are the system series independent modules.

Reliability Equations for 4-2 to 4-4 are as follows:

The reliability Equation for Figure 4-2 is Equation 4.5 below.

$$R_i = 1 - \prod_{j=1}^{n_i} \left(1 - R_{i,}^{j}\right) \qquad 4.5$$

The reliability Equation for Figure 4-3 is the same as Equation 4.3 earlier discussed. Here, it is referred to as Equation 4.6 for the purpose of numbering orderliness.

$$R_i = 1 - \prod_{m=1}^{n_i} \left[\prod_{j=1}^{x_i} \left(1 - R_{i,m}^{j}\right) \right] \qquad (4.6)$$

The reliability Equation for Figure 4-4 is below:

$$R_s = 1 - \left[\prod_{m=1}^{n_i} \left(1 - \prod_{j=1}^{x_i} \left(1 - R_{i,m}^{j}\right) \right) \right] \qquad 4.7$$

4.4 Chapter Summary

In summary, redundancy allocation problems were traditionally implemented on systems sequentially. In this chapter redundancy allocation problems are extensively explored and analysed at all levels, with specific mathematical formulations adopted for reliability calculations for each system level.

The author demonstrated the feasibility of allocating redundancy beyond the usual traditional approach to an integrated approach. The newly established approach which showed no limitation on either levels (i.e., components, module, branch, sub-branch and system) was actualised in *MATLAB* software tool. In the following chapter five, to be specific, the author provides a user friendly manual to explain both the operations and functionalities of the *MATLAB* software tool. The chapter also features a case study detailing how a series-parallel system was configured in line with a modern way of coding for each system level. A three dimensional Pareto frontier results with the respective project objectives such as cost, failure rate and weight are also shown and analysed in each of the dimensions.

5. Software Development

This developed software program was achieved in *MATLAB* environment, and stored in a main directory called Multi-level, Multi-Objective Redundancy optimisation (*MLMOR2*). The program is developed to run very fine in all the case studies used in this project having the correct data. Therefore, at first, when the user loads the main directory it sequentially locates other directories whose individual functions vary from one another but are collectively channelled towards delivering the program objectives;

- Pareto front analysis with cost, weight and FR as objectives
- Integrated redundancy allocation, in which two cases dealing with different topology level are shown and analysed in Figures 5-7 and 5-8.

5.1 Program Formulation

The formulated *MATLAB* program was achieved with respect to its objectives. It was subsequently portioned by creating directories for each program milestone. Some tasks did require simultaneous approach. Directories created, when loaded by the main directory, locate each other sequentially. This led to consistent debugging practice on each directory in order to have the program run successfully. The entire directories created for this program, as well as the activities of each of these directories, are detailed in the modular structure of the software shown in Figure 5-1 below.

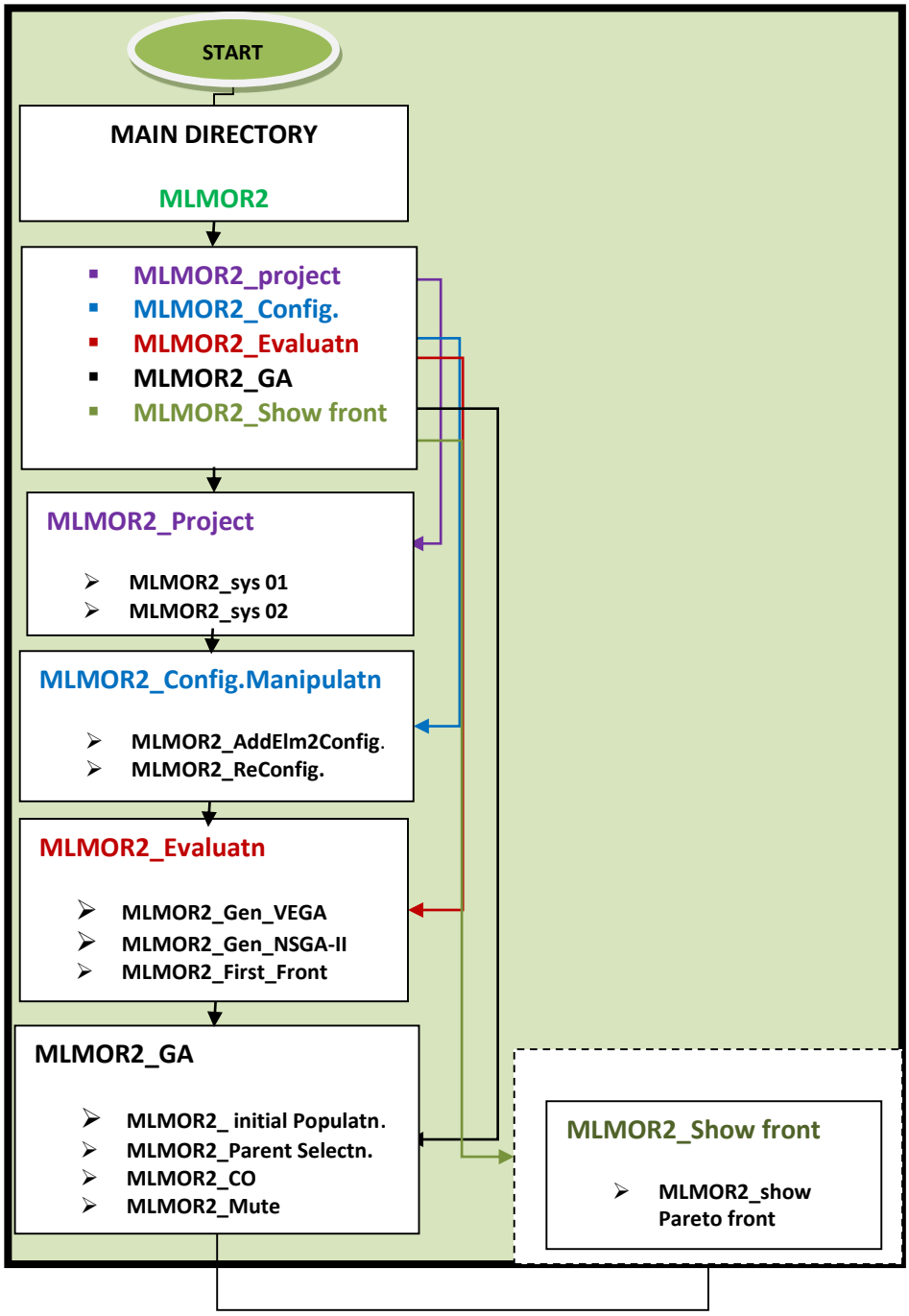


Figure 5-1 Modular structure of the software.

5.2 Main Directory (MLMOR2)

The main directory contains other directories created for this program. Therefore, when the user loads the main directory; it sequentially locates (*MLMOR2_project*) as one of the sub directories contained in the main directory.

5.2.1 MLMOR2_Project

Upon location of the *MLMOR2_project*, it does the following;

- Sets *GA* optimisation parameters (i.e., *npop, ngen, pc and pm*)
- Sets the generated initial population
- Produces results in Pareto frontier through efficient application of two multi-objective solution techniques vector evaluated genetic algorithm and non-dominated sorted genetic algorithm (*VEGA* and *NSGA-II*). Upon the solution, the decision maker (*DM*) can then make an informed decision in accordance with the prioritised objective(s); cost, weight and failure rate (*FR*). In this case, they can either be single or multi-objective. Refer to Figures 5-2 and 5-3 for the solutions.

5.2.2 MLMOR2_ConfigurationManipulation

When this directory is sequentially located, it automatically does the following;

- Sets the system configuration parameters such as the entire components availability meant for each modules at various system levels.
- Selects and sets the minimum and maximum components for each module from the list of component availability.
- Allocates the exact required redundancies for each of the system level:
  - (i) Module
  - (ii) Branch
  - (iii) Sub-branch without limitation.
- Updates the location of each system branch in the simplified configuration. Please refer to Figures 5-7 and 5-8 for the solutions showing the overall system topologies in two cases.

5.2.3 MLMOR2_Evaluation

The evaluation directory evaluates and sets the *NSGA-II* and *VEGA* selection techniques adopted. Whilst the VEGA technique sets the selection process based on roulette wheel, the

NSGA-II techniques sets the selection based on tournament and enhances a better spread of solutions. Therefore, when the user loads the evaluation directory, it does the following.

- Sorts the index of the optimisation solutions accordingly (i.e., either in ascending or descending other in readiness for good solution output)
- Sets the different ranks in the population indicating different fronts where each of the objectives will be displayed for adequate decision making. In this case, the objectives are failure rate, cost and weight.
- Sets the defined crowding distances for each solution such that the solutions are well spread for analysis without overlapping.

5.2.4 MLMOR2_GA

The GA directory does the following;

- Generates and sets the initial population for the multi-level, multi-objective redundancy optimisation.
- Sets the defined new methods of chromosome definition that enables combination of building blocks at various system levels
- Generates the *VEGA* and *NSGA-II* selection parameters
- Sets the mutation and crossover configurations.

5.2.5 MLMOR2_Show Front

This directory does the following:

- Sets the solutions in three dimensions in the Pareto front.
- Sets each of the objectives in one of the dimensions in the Pareto front as depicted in Figures 5-2 and 5-3.

5.2.6 Results Analysis of FR, Weight and Cost objectives

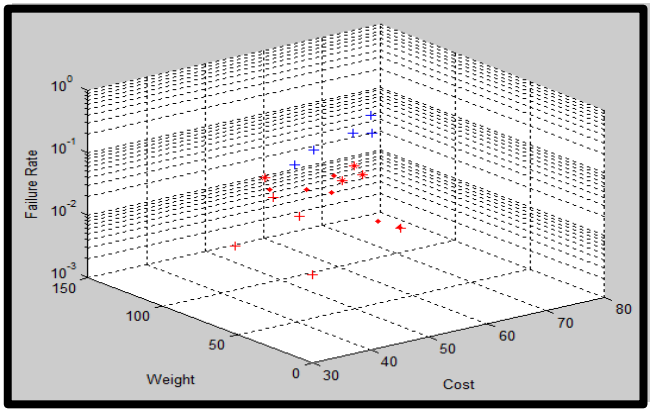


Figure 5-2 First Pareto front multi-objective solutions in three dimensions (3-D)

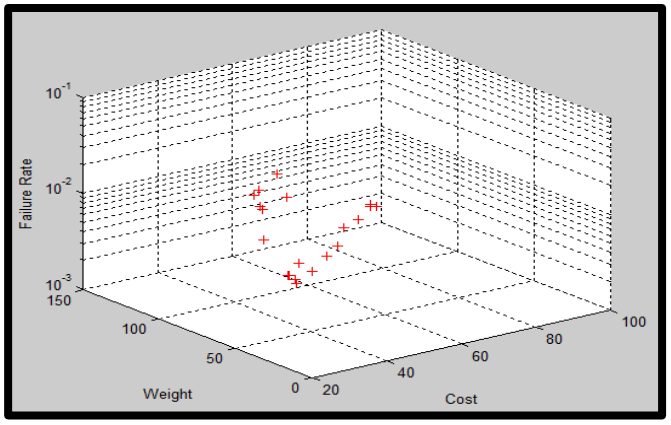


Figure 5-3 Second Pareto multi-objective solutions in three dimensions (3-D)

These results above are the product of the configured system discussed earlier in Figure 4-4, and are now displayed in Pareto front. The red and blue dots which appears in both cross and star signs represents the objectives (FR, cost and weight) solution. Both the first and the second Pareto fronts has the same objectives function. However, for the first Pareto front, the minimum and maximum cost are different from the second Pareto front. Whilst the first Pareto front cost range is measured at interval of ten, the second Pareto front cost range is measured at twenty. All the solutions are considered good, but the best solution point on the Pareto front depends on which objective is given priority. Since the system objectives are carefully taken into consideration during the design stage and met the design goals, it is imperative that the decision maker, where necessary, collaborates with relevant stakeholders to identify and prioritise

criteria associated with design goal. Priority can be to have a minimum FR of the system, minimum system cost and maximum system weight. Alternatively, priority may be to have the weight and failure rate of the system minimised. These solutions further zoomed increasingly in order to have two dimensional Pareto front for proper analysis of two objectives as showed in Figures 5-4, 5-5 and 5-6.

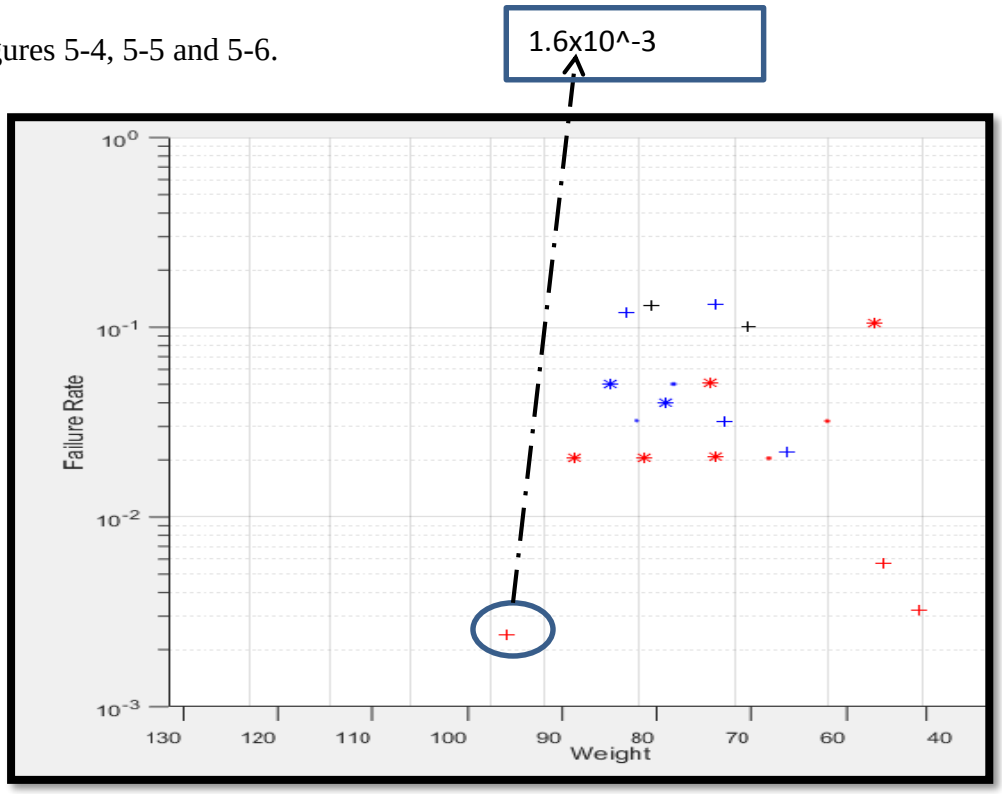


Figure 5-4 Pareto front showing best FR solution in 2-D

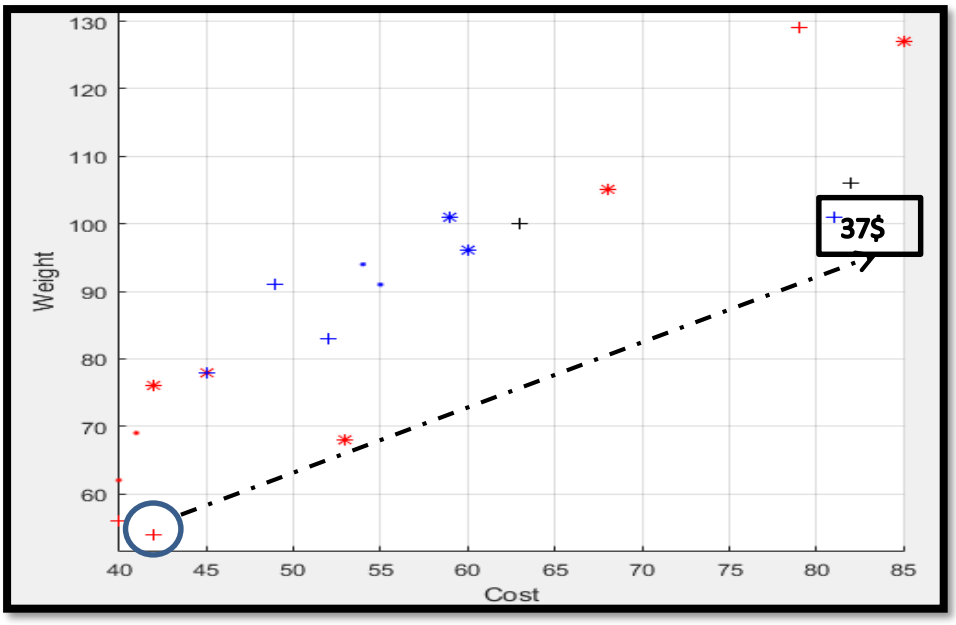


Figure 5-5 Pareto front showing best cost solution in 2-D

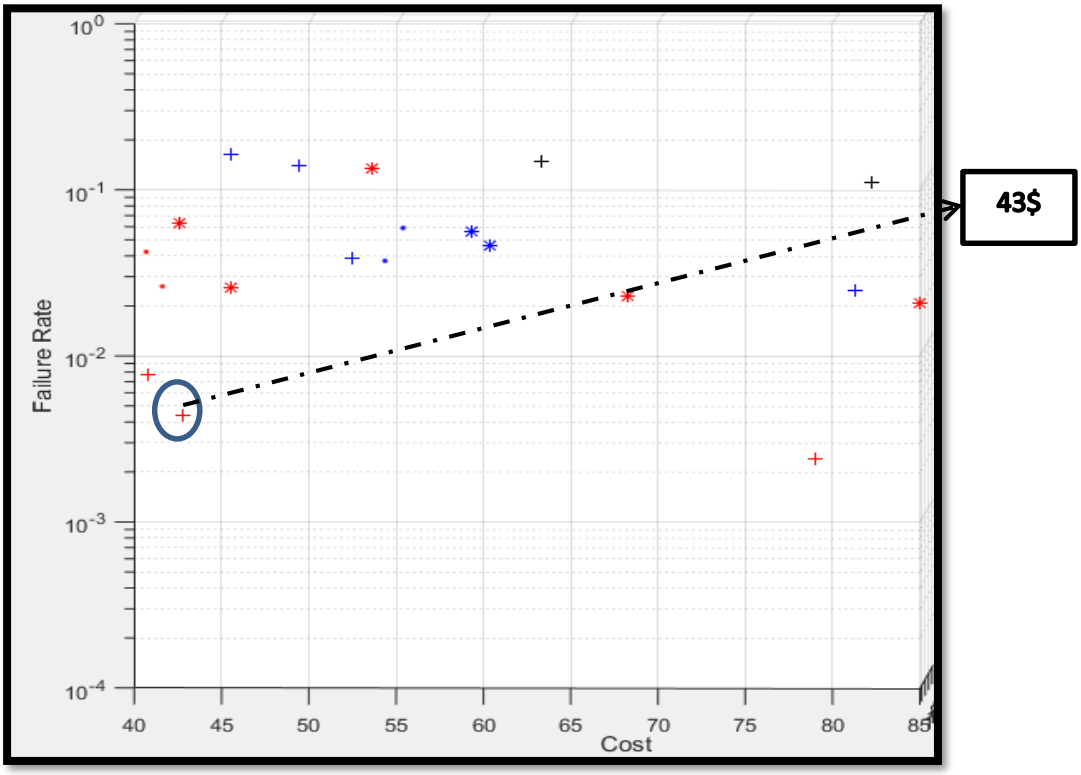


Figure 5-6 Pareto front showing best cost solution in 2-D

5.2.7 Generated System Topology of Cases I and II

5.2.7.1 System Topology Case I:

x17+((x3*x4+x8)*(x16+(x23+x14)+x12*x13))*((x3*x4+x8)*(x15+(x23+x14)+x13))+x20

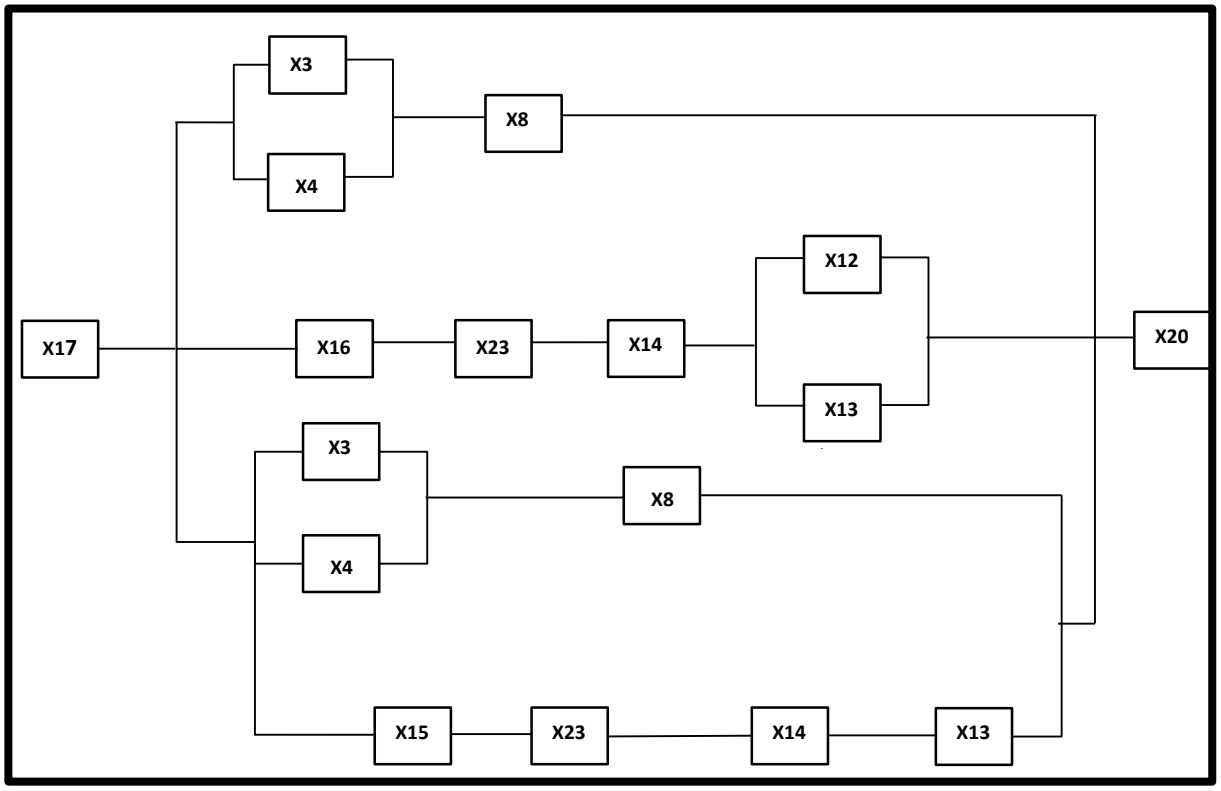


Figure 5-7 Case I: generic topology of the system

Result Analysis of case I.

This case represents a realistic integrated redundancies allocated, and the topology of the system under integrated redundancy allocation. As shown in Figure 5-7 above the result is a product of the system configuration discussed earlier in Figure 4-4 which was implemented in the *MATLAB* environment. From the result, it is a series-parallel system comprising branches, sub-branch, modules and components. The system has three branches parallel to each other. One of the branches, the third (lower branch) to be precise, has a sub-branch. In overall, each of the system branches has modules with parallel components (i.e., comprising upper and lower levels) to increase branch reliability.

The systems first and last modules with components are x17 and x20. The top branch comprises three modules consisting of components (x3, x4 and x8). Modules with components x3 and x4 are in parallel to each other, but in series to module with component x8. The middle branch comprises of five modules consisting of components (x16, x23, x14, x12 and x13). Module with components x12 and x13 are in parallel to each other, and have higher reliability. This is because it will require both components to fail before the module fails. While modules with components x16, x23 and x14 are in series, in overall the branch is connected in series, it therefore fails if any of the branch modules fail. The lower branch has a sub-branch. The upper branch has (x3, x4 and x8) modules. Module with component x8 is less reliable and in series to more reliable modules with components x3 and x4. The lower branch has modules consisting of components (x15, x23, x14 and x13) which are in series to one another. The upper and lower modules with their corresponding components that make up the system lower branch are parallel to each other, thereby making the branch very reliable. The branch fails if all the redundancies created in both upper and lower branches fail. From the Figure, it can be concluded that the system is reliable because it has good number of redundancies presents in all the branches and can perform efficiently.

5.2.7.2 System Topology Case II:

x18*x18+((x1*x1*x1*x2+x9*x6*x7)*(x15*x15+(x23+x14)+x12*x10))*((x4*x2*x4*x1*x3 +x8*x7*x8)*(x15*x16+(x21*x21+x14)+x12*x12))+x20

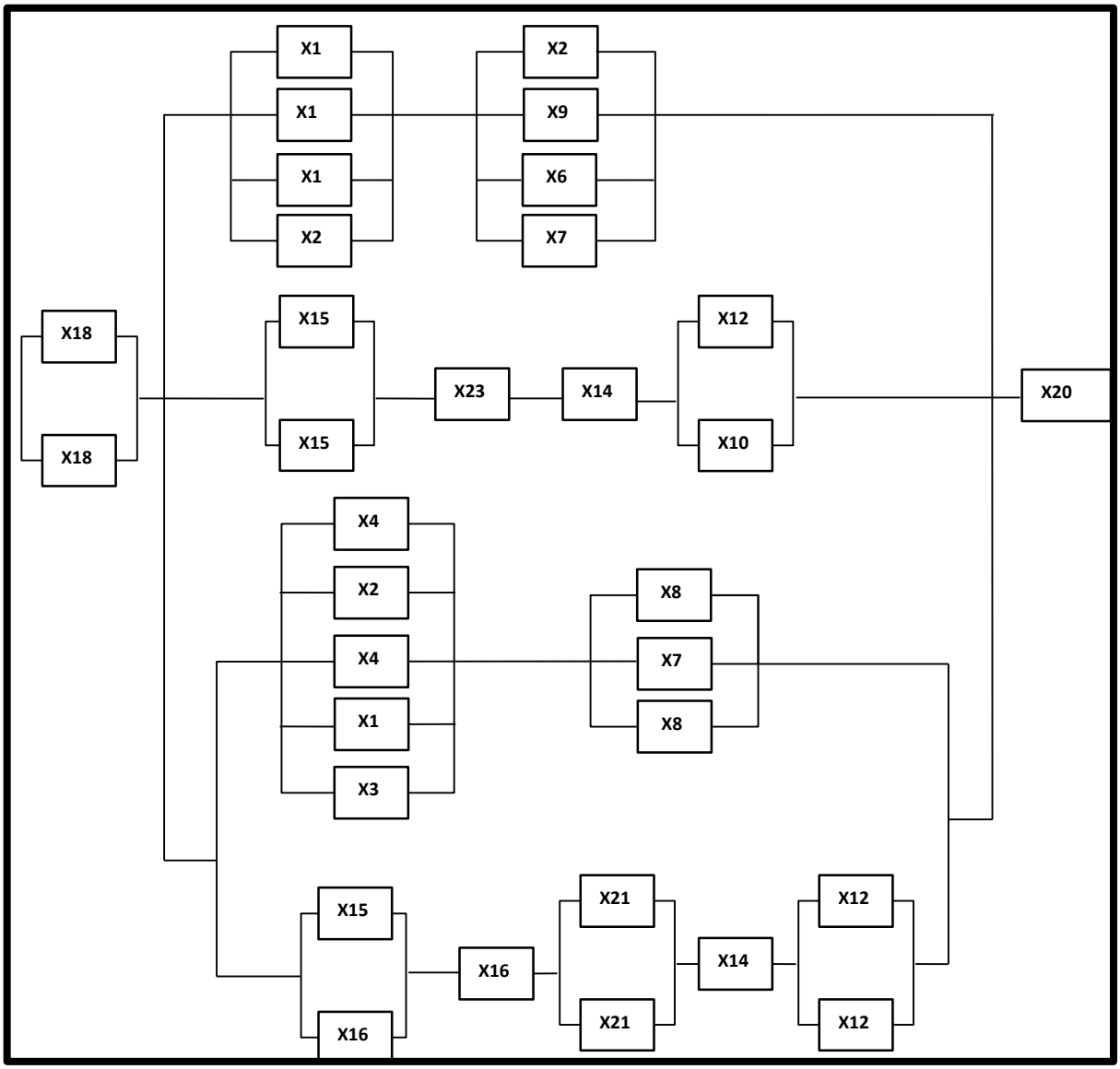


Figure 5-8 Case II: generic topology of the system

For this case, it also represents a realistic integrated redundancies allocated, and the topology of the system under integrated redundancy allocation. As shown in Figure 5-8 above, case II has a higher topology as compared to case I due to presence of more redundancies allocated on the system branches. However, both cases I and II, are product of the same system configuration discussed earlier in Figure 4-4 which was implemented in the *MATLAB* environment. Again, case II is a Series-parallel system comprising of full system branches and a sub-branch located in the lower branch as shown in the architectural topology. From the Figure, modules (x18 and x18) which are in parallel to each other, but in series to the entire

system branches, and the system terminates with an independent series module consisting of component (x20).

From the top system branch, redundant modules (x1, x1, x1, x2, x2, x9, x6, and x7) are allocated in parallel. However, modules consisting of components (x1, x1, x1, x2) and (x2, x9, x6, x7) each are in series to each other.

From the system middle branch, modules with two identical components (x15 and x15) are in parallel to each other, modules with components (x23 and x14) are in series. Modules consisting two un-identical components (x12 and x10) are again in parallel to each other. However, all of these modules with their components allocated to this branch are all in series to the system middle branch.

The system lower branch is made up of an upper and lower branches. From the upper branch, five modules consisting of components (x4, x2, x4, x1 and x3) are separately in parallel. This same scenario is applicable to module with components (x8, x7 and x8) in the same upper branch. These components are connected in parallel to provide high reliability to the system branch. From its lower branch, two un-identical components (x15 and x16) are in parallel to each other. Also, modules with identical components such as (x21 and x21) and (x12 and x12) are connected in parallel to each other, while modules with un-identical components (x16 and x14) are connected in series. In comparison, Case II has higher topology and higher reliability levels than case I due to presence of strategic redundancies created and can outperform case I.

5.3 Chapter Summary

This chapter presents the usability of the *MATLAB* software tool used in this work with its functionalities explained. The author did show that it is a prolific software tool for technical project delivery. From the project conceptual stage, particularly projects involving Series-Parallel systems, the tool enabled the designer to configure design candidates in accordance with specification to produce the desired output. In this case, desired outputs were shown in two phases:

- Showing system topology levels with integrated allocated redundant components which outweighs the traditional design approach on Redundancy allocation problem
- Showing the multi-objective results on the Pareto frontier for decision making.

The program works efficiently well, especially as it only requires minimum run time. This method of system configuration using *MATLAB* software tool can be adopted in various engineering fields to optimise or backup a system for reliability purposes. In fact, in the following chapter Six, this successful *MATLAB* software tool is also used for the configuration analysis of case studies.

6. Case Studies

This chapter addresses couple of the thesis case studies and compared the results of the proposed approach with other approaches mostly adopted in optimal redundancy allocation practice. The systems under consideration are multilevel system and a fuel oil system. Prior to this work, no Suitable codes representing the various systems were provided. However, apart from comparing the optimal solutions of the multilevel system, suitable codes for the various systems were provided. With respect to the oil fuel pump system, the proposed method specifically targeted to make the system more redundant, and provided suitable codes representing the new redundant fuel oil system. Figure 6-1 below is a series system showing maximum redundancy configuration for U_1 consisting three blocks U_{11}, U_{12} and U_{13} at the second level.

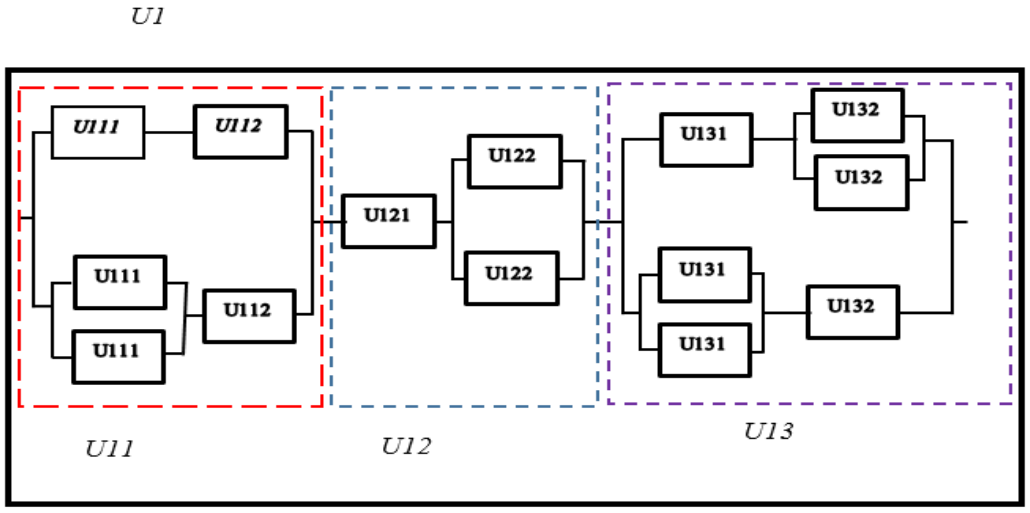


Figure 6-1 Multilevel reliability system

The system level can further be configured without limitation. To achieve this, the author first renamed the system from U_1 to System A, alongside with its blocks from ($U_{11} - U_{132}$) to $(A1 - A10)$ without altering the original configuration. The reality of this new name is shown in the Figure overleaf:

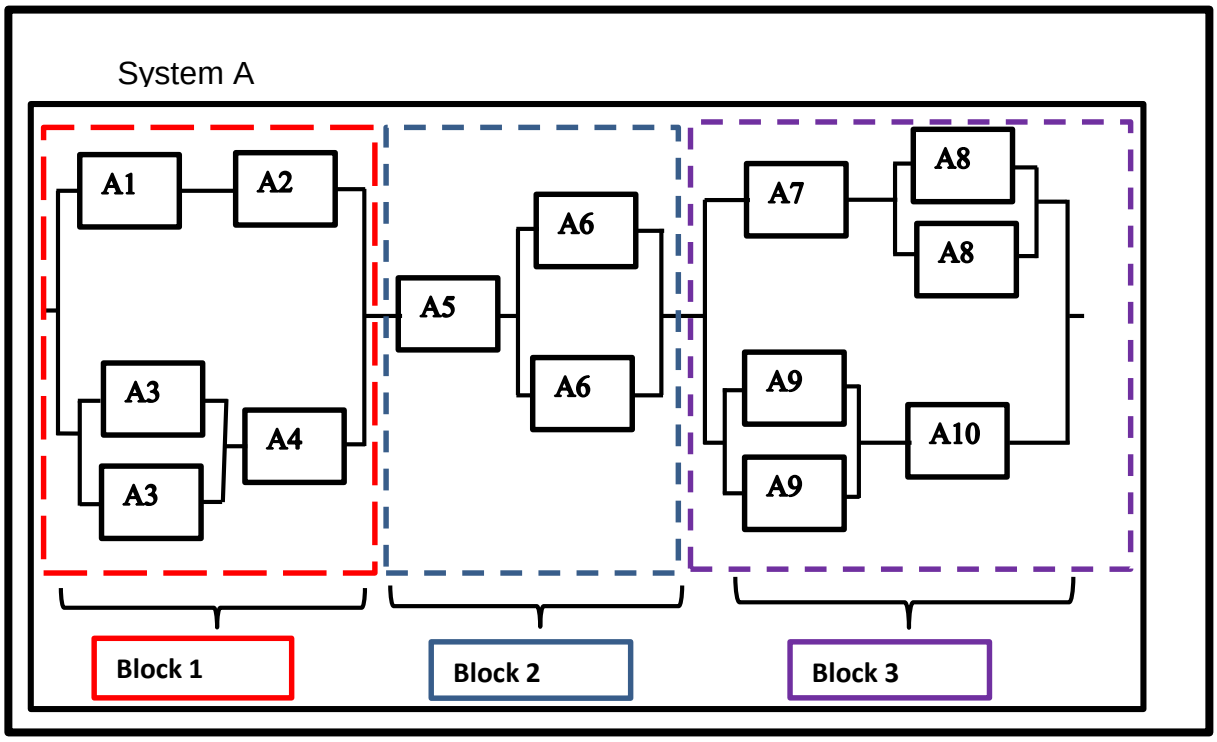


Figure 6-2 Renamed multi-level reliability system of Figure 6-1

Figure 6-2 is subsequently codified in *MATLAB* programming environment to show the solution of this system in code version. One of the advantages of a codified solution is that it is quite flexible to adopt to improve system levels without limitation. *MATLAB* configuration helps not only in defining the exact number of components a block should have, but also specifying the number of branches that exist in each block. In fact, it is a more convenient approach to design and redesign a system to any level. Because *MATLAB* is an effective design tool and several programs with different approaches can be implemented, it is therefore important to specify that *NSGA-II* is applied to codify solutions of these blocks.

The configuration showed the possibility of a system block having un-identical components in series but uses identical components in parallel to increase the block required maximum level, as depicted in Figure 6-3.

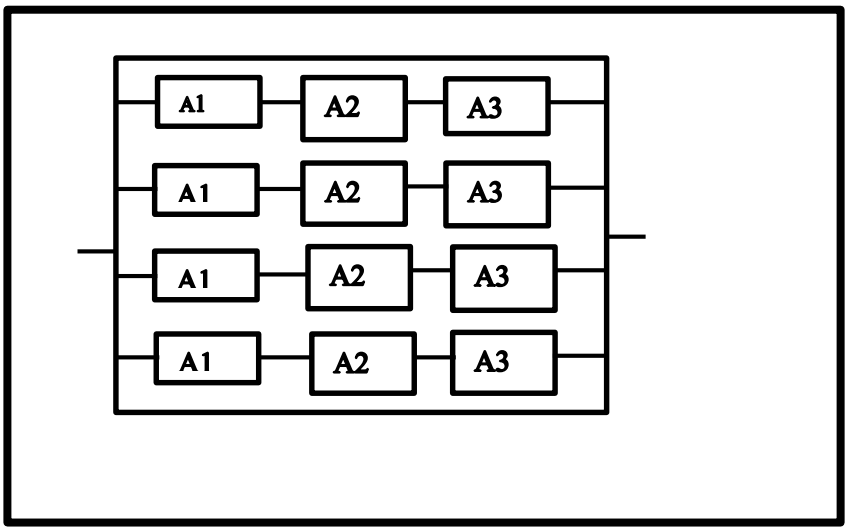


Figure 6-3 Maximum redundancy level using GA

Figure 6-3 excludes from consideration a suitable code capable of improving the block levels to any level of choice (Multi-level) without losing performance efficiency. Systems with multi-level have better performance efficiency because of the wider feasible design space it occupies. The more design space a system has, the more efficient it performs. *GA* do not have much feasible design space because it reduces feasible design space in the course of transforming hierarchical design variables into a one dimensional array. In addition, the development can have an adverse effect on the *GA* overall performance such that *GA* may fail to get superior solutions that exist just outside its feasible design space.

GA configuration leads to performance deficiency as a result of the feasible design space restriction. However, such a restriction problem was to a certain extend addressed using Hierarchical Genetic Algorithm (*HGA*). *HGA* provided a better feasible design that permits possible combination of multi-level redundancy allocation. This is contrary with the *GA* configuration scheme that does not allow redundancy at two levels simultaneously as evident in Figure 6-3. *HGA* configuration scheme allows redundancy at two levels simultaneously as

shown in Figure 6-4, and therefore has the tendency to outperform conventional *GA* considering it has more design space to yield a better solution.

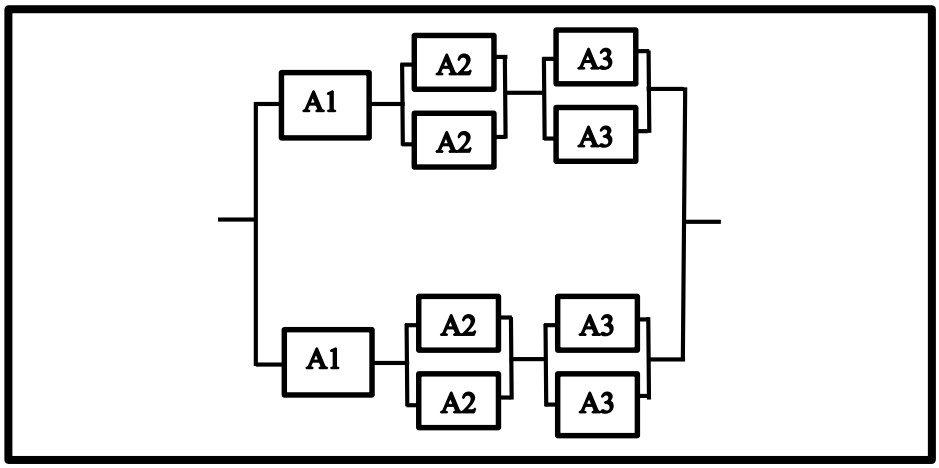


Figure 6-4 Maximum redundancy level using HGA

One of the shortcomings with HGA configuration is the lack of flexibility to increase any part of the system levels beyond allowing redundancy at two levels simultaneously, which is one of the novelties achieved in this project using *NSGA-II*.

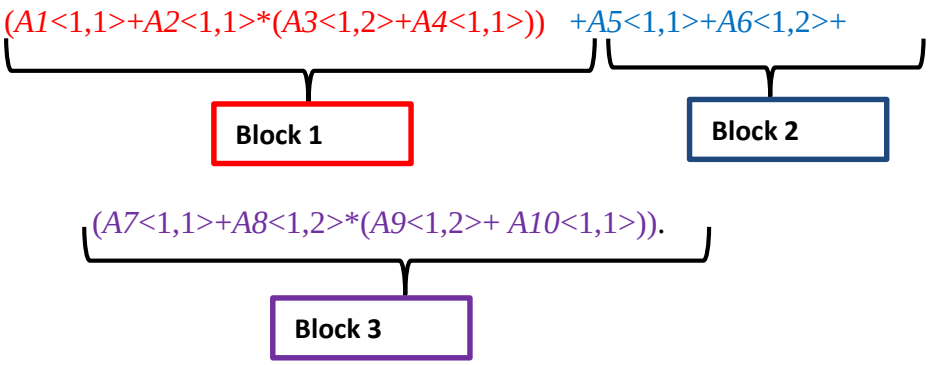
6.1 NSGA-II Code Solutions of the above Configuration Obtained Earlier with GA and HGA.

Codified solution versions of Figures 6-2, 6-3 and 6-4 respectively are presented in the form of cases. Each case clearly handled each Figure obtained from the case study and is therefore presented accordingly.

Case 1: codified solution of Figure 6-2.

Figure 6-2 is made up of three blocks with redundant modules (*A1-A10*).

Block 1 has redundant modules (*A1-A4*), Block 2 has redundant modules (*A5-A6*) and Block 3 has the remaining redundant modules (*A7-A10*). Codes for each of these blocks are shown overleaf:



6.2 Code Solution Analysis of Figure 6-2

The code accordingly defined exact redundant modules belonging to each block alongside their branches. For block 1, the code showed that four modules (*A1- A4*) and two branches (upper and lower) exist. The upper branch has modules *A1-A2* linked in series and each of the modules can only have one component, while the lower branch also has redundant modules *A3-A4*. The modules located at block 1 lower branch are more redundant because module A3 can have a maximum of two components in parallel while the *A4* module is restricted to have one single component. It therefore requires both components in module A3 to fail simultaneously for the module to be in a failed state.

For block 2, there are only two redundant modules (*A5-A6*). Module *A5* can only have one single component while module *A6* can have two components maximum, and these components can be visibly seen in parallel in block 2.

Block 3 has two branches as clearly shown in the configuration with a total number of four modules (*A7-A10*) in the upper and lower branches. It will not be out of order to agree that the branches have equal reliability strength considering that both the lower level and the higher level branches have one module each in parallel. The upper modules of the block are *A7-A8*.

The configuration showed that *A7* can neither have more than one component nor can *A8* have more than two components. This is applicable to the lower branch of the system of the block with modules (*A9-A10*). Module *A9* can only have two parallel components maximum, while module *A10* cannot have more than one component.

The entire system fails if either of the blocks fails completely, because series system blocks are dependent on the other in order to function.

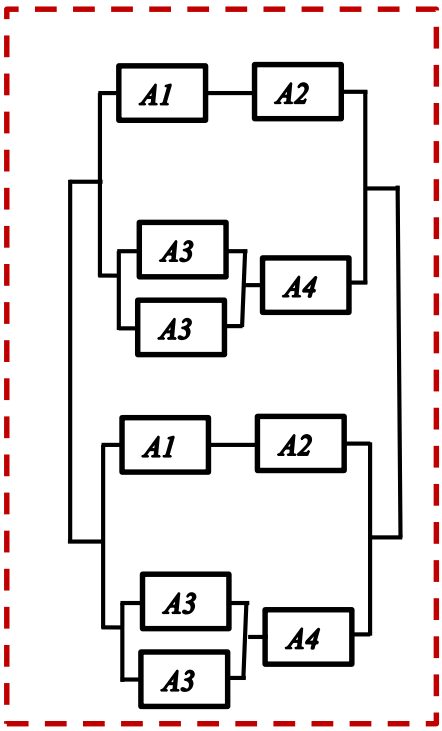
As stated earlier, the system level can be expanded to multi-level without limitation to all of the system levels. Block 1 can be expanded to any level of choice through technical modification of the system code. For example, new block 1 configuration that is aimed at only expanding the block branch to a new level of choice becomes;

(*A1*<1, 1>+A2<1,1>*(*A3*<1,2>+*A4*<1,1>))<1,*n*>.

New Block 1 code

The solution to this code when the parameter *n =2* is presented in Figure 7-5

The parameters such as *m, n and p*, used to technically modify the codes, represent the number of levels desired. This can be from the components levels down to the system levels. Therefore, they vary in terms of numbers from one to infinity. In other words, the parameters vary from a single level to multi-level.



⟷ (*A1*<1, 1>+*A2*<1,1>*(*A3*<1,2>+*A4*<1,1>))<1,2>.

Figure 6-5 Expanded block 1 level when Parameter *n =2*

Case 2: codified solution of Figure 6-3

(*A1*<1,1>+*A2*<1,1>*A3*<1,1>)<*1,n*>.

As earlier stated, parameter n is the maximum number of redundant branches the block can attain. If parameter *n =10,* then the block level rises from the present single level to the specified level ten which will actually yield better reliability performance.

Case 3: codified solution of Figure 6-4

(*A1*<1,1>+*A2*<1,m>+*A3*<1,n>)<1,p>. A typical solution to this code is Figure 6-4 when the design parameters *m=2, n=2 and p=2*.

Case 4: codified solution of MLMOR shown in Figure 6-6 overleaf.

((A1<1,1>+A2<1,m>+A3<1,n>)<1,p>(A1<1,1>+A2<1,1>+A3<1,1>)*(A1<1,1>*

+A2<1,1>+A3<1,1>))

Thus, solution to the code is shown below and it can be consistently applied to improve the system to any level. In addition, it can express the exact internal and external structure with both series and parallel linkages.

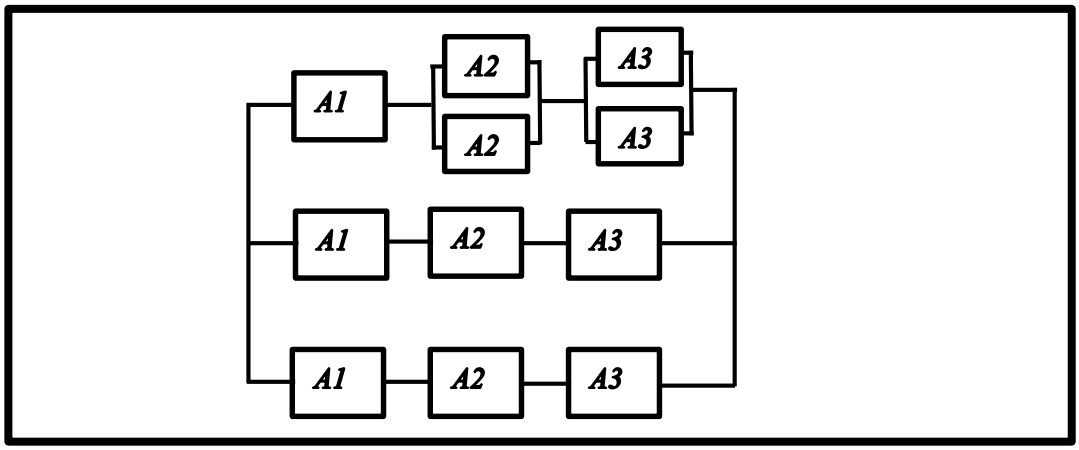


Figure 6-6 Solution of MLMOR using NSGA-II

The solution showed branches (lowest, middle and upper) present in the block and each of these block levels can separately be improved without causing adverse effect on other block levels. This is because *NSGA-II* does not need vector transformation, the feasible design space is normally not affected, and this leads to better reliability performance. From the solution, the block FR has been handled in a manner that results into effective reliability performance on the block. Interestingly, the improved reliability gotten using *NSGA-II* code was achieved without incurring any additional material or components costs. In higher reliability applications, this is a fundamental milestone because even very small improvements in system reliability are usually difficult to achieve.

NSGA-II was subsequently applied to optimize multilevel series redundancy allocation problems with distinct configurations referred to as problem A and problem B respectively as shown below in Figures 6-7 and 6-8. These problems have levels, problem A contains three levels whilst problem B contains four levels.

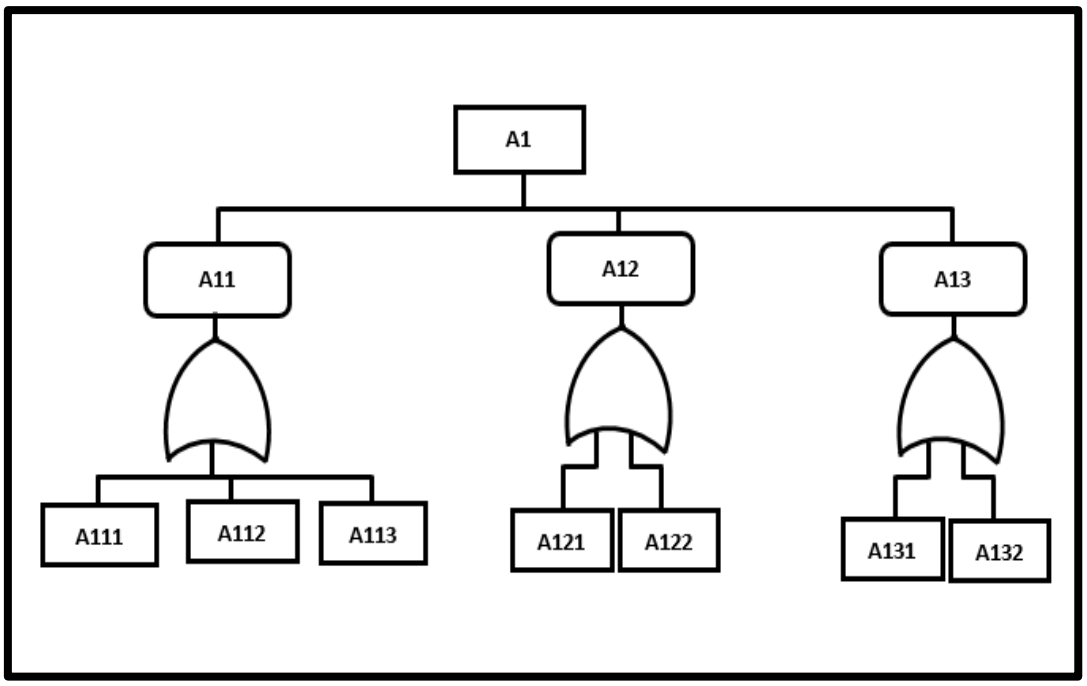


Figure 6-7 Problem A: a three multilevel system

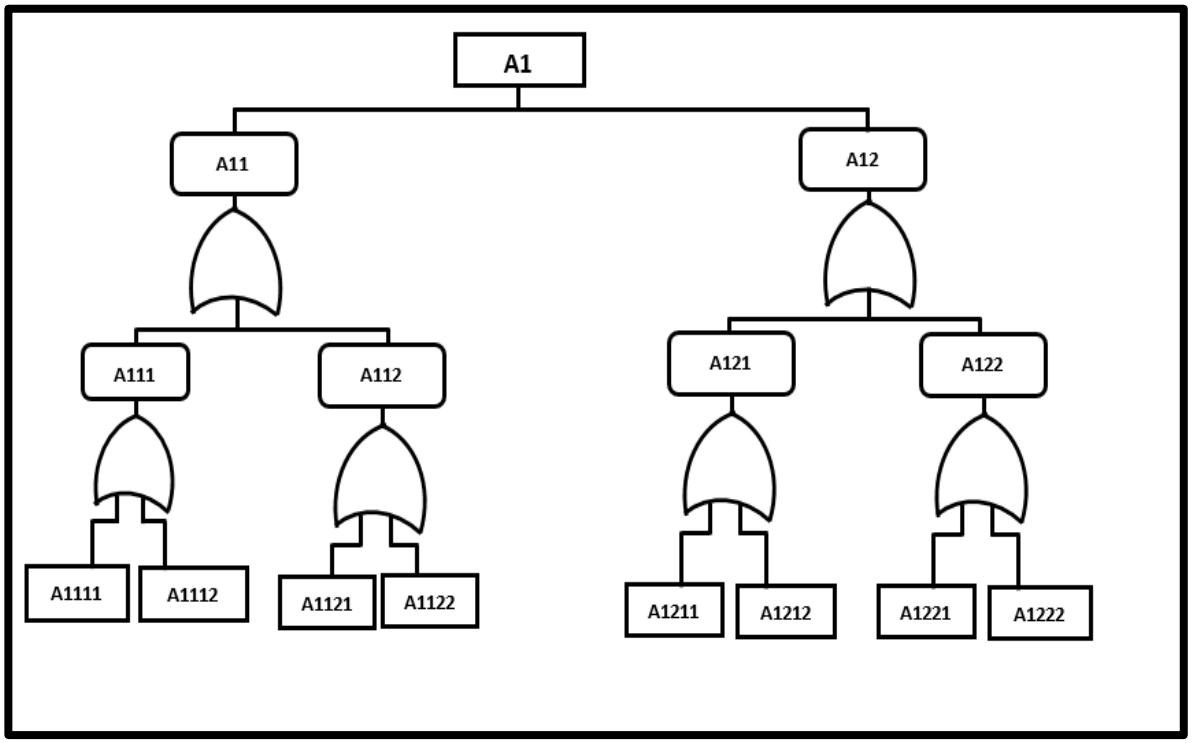


Figure 6-8 Problem-B: a four multilevel system

The candidate compared the *NSGA-II* result obtained with that of convectional *GA* and *HGA* to show that *NSGA-II* performance is better, especially with respect to assessing influence of cost constraints upon optimal solutions as shown in Figures 6-9 and 6-10. These Figures demonstrate that the optimal solution produced with the proposed *NSGA-II* approach is far better than the solutions produced by both conventional *GA* and Hierarchal *GA* under the same computational environment.

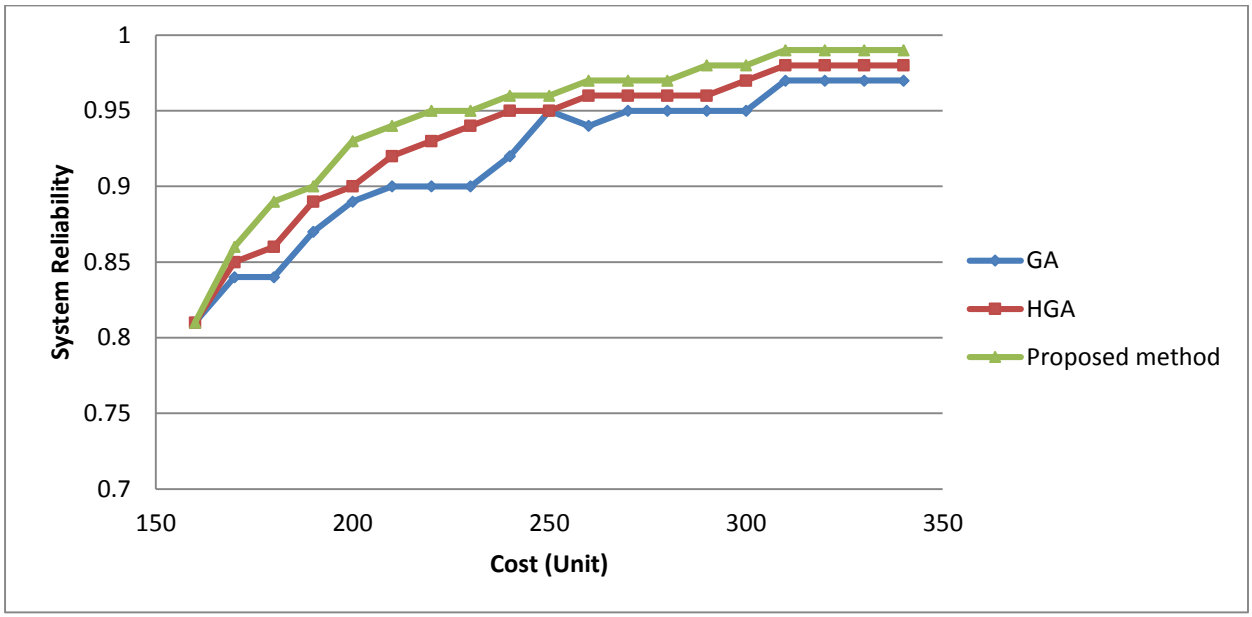


Figure 6-9 Optimal Solution for problem – A obtained using GA, HGA and NSGA-II (proposed method).

Twenty (19) cases were examined for Problem A (a 3-level) in order to assess the influence of cost constraints upon optimal solution. Ten 400 generation trials were performed using each algorithm type to trace the best solution in each of these ten-trail set, and the best solution of the ten-trial set was chosen as the optimal solution in each of these cases as shown in Figure 6-9. In like manner as above, 15 cases for problem B (a 4-level) system was also examined for the same purpose of assessing the influence of cost constraint upon optimal solution, using

again ten 400 generation trials. The below Figure depicts best optimal solution chosen for the ten cases considered.

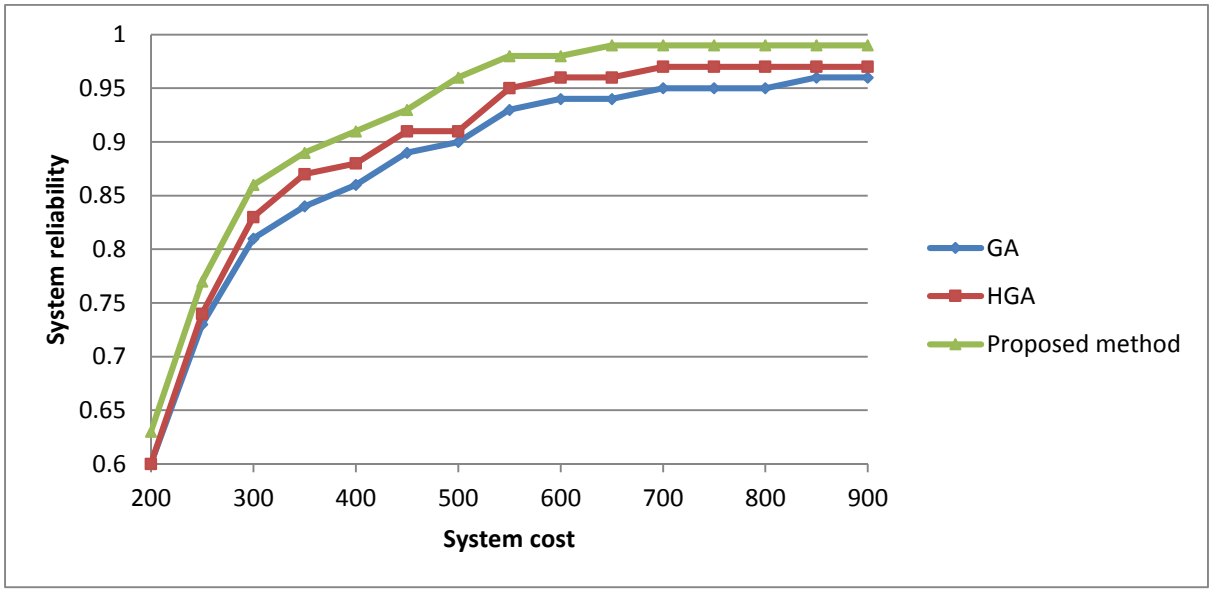


Figure 6-10 Optimal Solution for problem – B obtained using GA, HGA and NSGA-II.

RBD showing best GA, HGA and NSGA-II solutions for problem A with three levels is shown in Figures 6-10.1, 6-10.2 and 6-10.3 respectively.

System A1

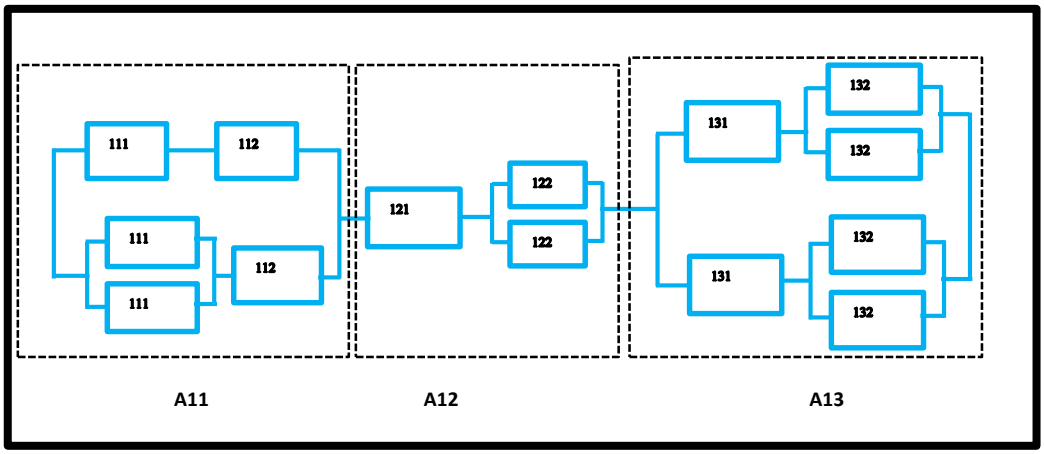


Figure 6-10.1 -Best solution of GA

System A1

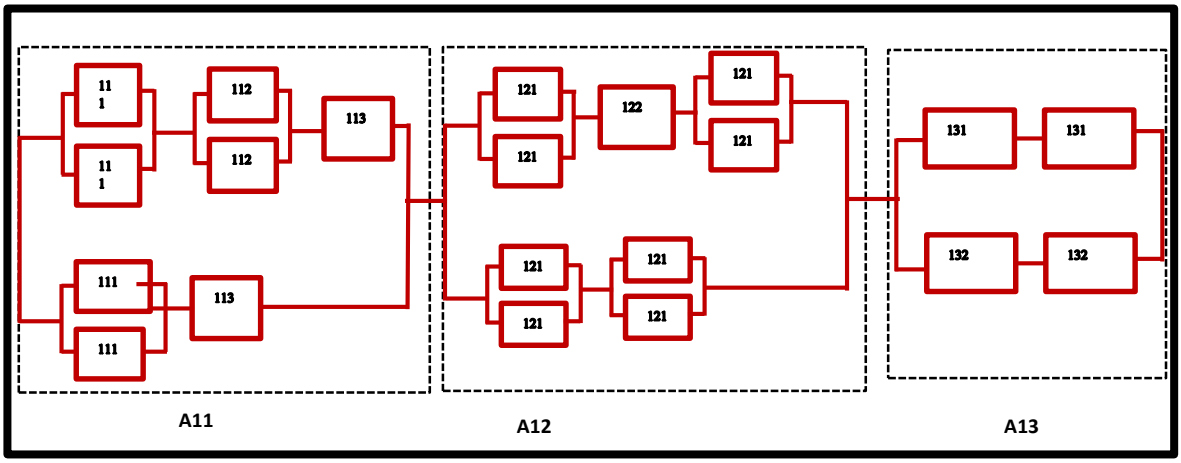


Figure 6-10.2 –Best solution of HGA

System A1

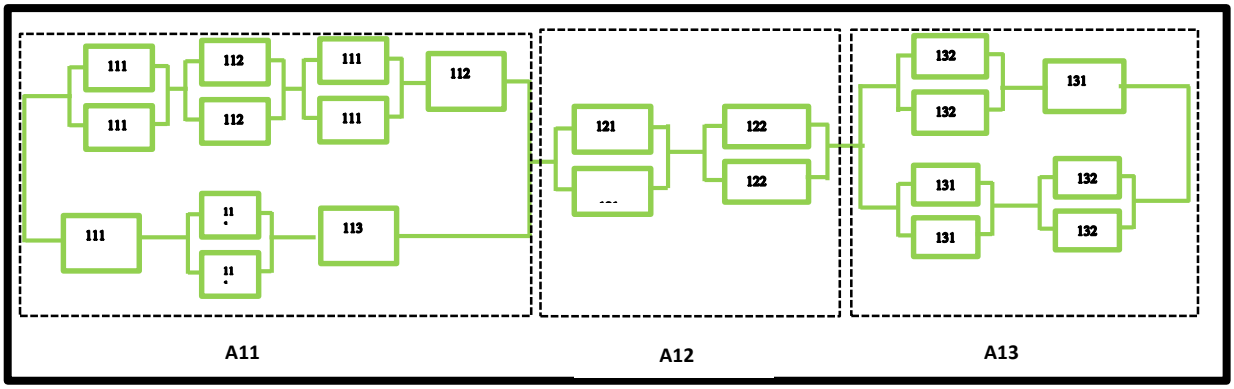


Figure 6-10.3 Best solution of the proposed method (NSGA-II).

RBD showing best GA, HGA and NSGA-II solutions for problem B with Four levels is shown below in Figures 6-10.4, 6-10.5 and 6-10.6 respectively.

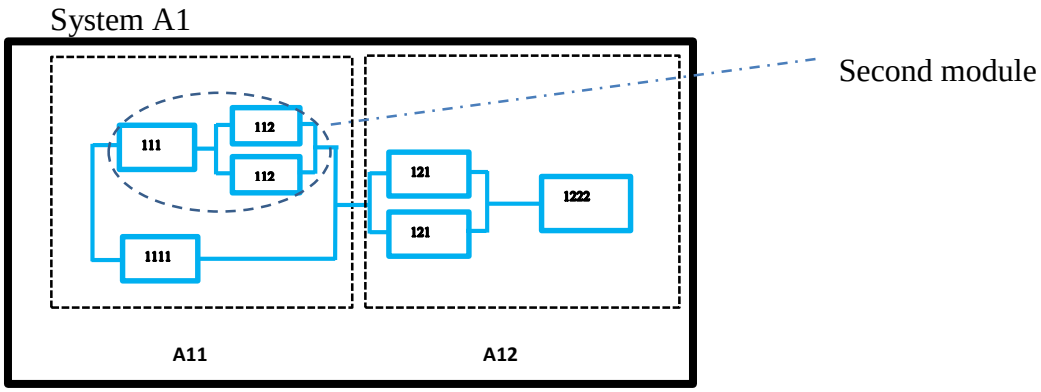


Figure 6-10.4 best solution of GA

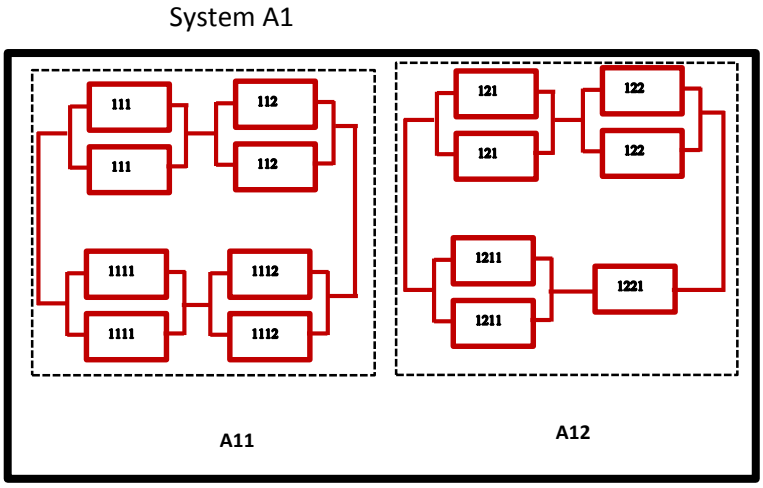


Figure 6-10.5 best solution of HGA

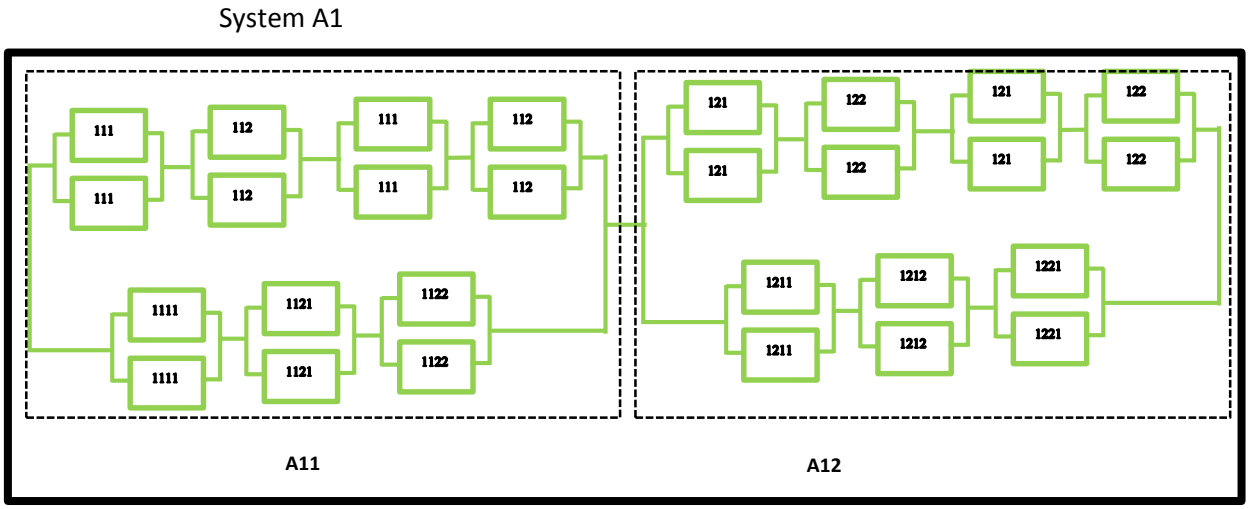


Figure 6-10.6 best solution of the proposed method (NSGA-II)

As explained earlier the RBD results shows that the proposed method can have more redundancies at two levels simultaneously which is why it has optimal reliability performance.

6.3 Fuel Oil System Case Study

To further demonstrate the application of design optimisation in practice, the author applied the optimisation capabilities of reliability block diagram alongside with this project written codes to a manually optimised fuel oil service system for a cargo ship, as shown in Figure 6-11

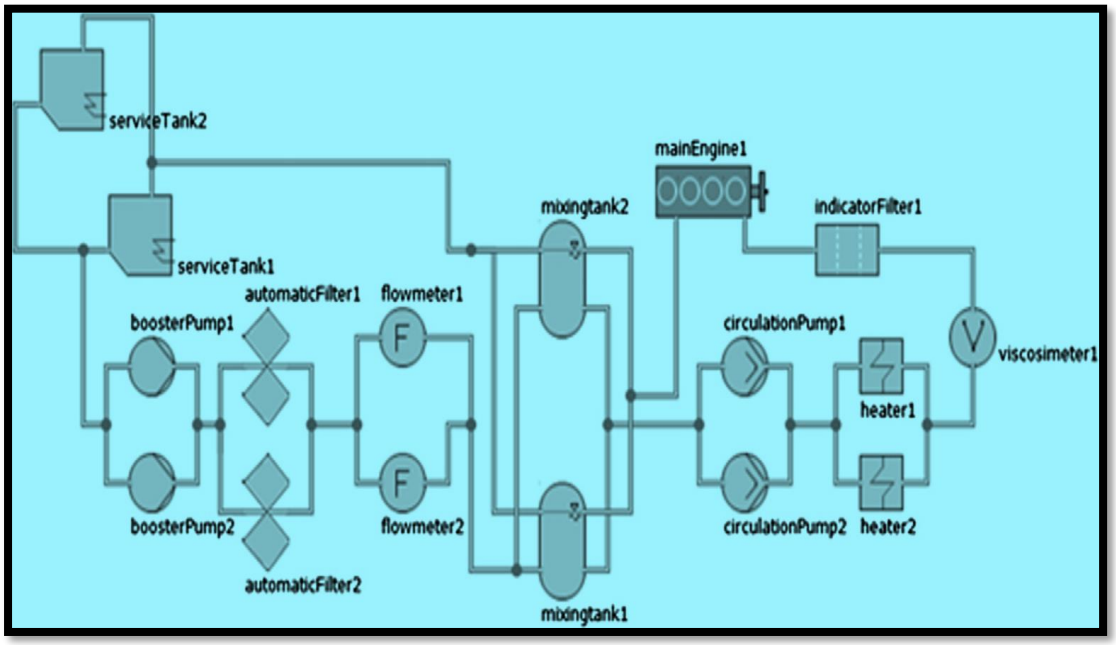


Figure 6-11 Fuel oil system-Manually optimised (Yiannis, P. *et al.*, 2011)

Oil is pumped to the system service tanks and flows through to the mixing tanks. The mixing tanks collect circulated oil and also acts as reserve tanks hence it supplies fuel when service tanks are empty. The fitted flow meters indicate fuel consumption while the booster pumps are responsible in pumping the oil through the heaters which heats the oil. The viscosimeters control the system fuel oil temperature in order to provide accurate viscosity for combustion. High pressured fuel is discharged through the circulation pumps. Indicator filters prevent particles in the oil from entering the engine. Whilst the main engine is usually operated in accordance with manufacturer's instructions to manoeuvre on heavy fuel oil, automatic filters provides full-flow fine filtration of heavy fuel oil. Since the objectives are to minimise the

system failure rate and increase the reliability through components replication on each of the subsystems, it is therefore unnecessary to further expatiate on the system functionalities. To realise the objectives will originally lead to a rise in the system topology level to a multilevel as against the base design with low minimal component replications as shown in Figure 6-11. Failure of this system leads to loss of engine propulsion and can possibly result in the ship becoming grounded as a result of drifting. To minimise this system failure rate, the candidate optimised the system as shown in Figure 6-12 and annotated the subsystems using A1-A10, to represent various components for easy system analysis. The system components replication did not consider the main engine component (A6) because the main engine is immutable. The remaining replication was done in accordance with the *NSGA-II* codes specification and can over time be validated to any level without restrictions.

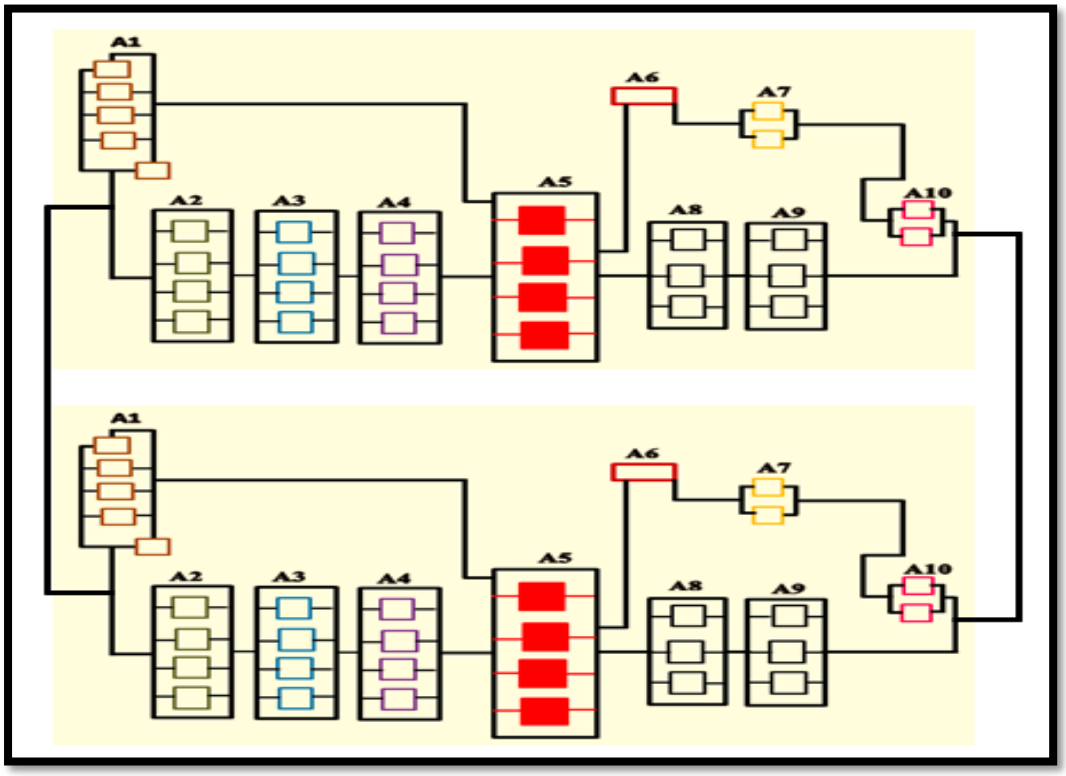


Figure 6-12 Fuel oil system-Optimised design using NSGA-II to Multilevel

Annotations: A1- Service tanks, A2- booster pumps, A3- automatic filters, A4- flow meters, A5-mixing tanks, A6-main engine (not replicated), A7-indicator filters, A8- circulation pumps, A9- heaters, A10- viscosimeters.

The code for the above system is shown and analyse below as:

'((A1<1,4>*A2<1,4>+A3<1,4>+A4<1,4>)*A5<1,4>*(A6<1,1>+A7<1,2>*A8<1,3>+A9<1,3>*A10<1,2>))<1,n>' Where parameter n=2.

After replicating the subsystems, the overall system was equally replicated to a higher level in order to perform optimally in case of failure on one branch. The two branches, upper and lower branches, provide optimal performance and are highly reliable due to additional parallel redundancies allocated appropriately on the subsystem. The code clearly specified the maximum allowable number of components each of the system's subsystems can have. For service tanks, booster pumps, automatic filters, flow meters and the mixing tanks the maximum allowable components is between one and four. For indicator filters and viscosimeters subsystems, the maximum allowable components each can accommodate is between one and two. For circulation pumps and heaters, the maximum allowable components for their respective subsystems are between one and three. Whilst the main engine was not replicated because, it is immutable as afore-said. The maximum topology level the system can attain in line with the code is two since parameter *n* in the code is between one and two. The system topology levels increases with increase in parameter n. In addition, the cost and FR of the components that make up the system in Figure 6-12 are shown in Table 6.1. The data was not made use of in this chapter as the author only targeted to create more redundancies on the system. However, the data can be employed for extensive cost and FR analysis of this system with respect to each alternative.

Table 6.1- Alternative cost and failure rate attributes for components of the fuel oil (Yiannis, P. et al., 2011).

	Alternative 1		**Alternative 2**		**Alternative 3**	
Components	**Cost**	**Failure Rate**	**Cost**	**Failure Rate**	**Cost**	**Failure Rate**
Indicator filter	1500	5.00E-07	2500	2.00E-07	3222	1.00E-07
Viscosimeter	2500	2.50E-07	3178	1.00E-06	3814	5.00E-07
Heater	2000	6.70E-07	2505	5.00E-06	3956	1.00E-06
Circulation ump	6000	3.20E-05	13380	2.00E-05	18000	7.00E-06
Mixing tank	2000	1.60E-05	2963	8.00E-06	4444	2.00E-06
Flow meter	2000	1.00E-05	3000	1.00E-06	4444	5.00E-07
Automatic filter	2000	1.00E-05	2647	5.00E-06	3529	1.00E-06
Booster pump	5000	3.20E-05	10682	2.00E-05	12500	5.00E-06
Service tank	1500	1.60E-05	1957	5.00E-06	2739	1.00E-06

6.4 Chapter Summary

In this chapter, the written NSGA-II codes were technically applied on two case studies involving a multilevel reliability system, and a fuel oil system, in order to further optimise and analyse these systems. For the fuel oil system, the objective was simply to minimised the system FR through allocation of more redundancies to the system components and modules. Afterwards, because the value of the parameter 'n' in the code, which represents branch level was 2, an additional system branch was achieved and this resulted in a high increase in the systems overall topology in accordance with code configuration validated against the fuel oil system. For the multilevel reliability system, the original system configuration was transformed to suit the research method of system configuration. Each of blocks (modules) in the system was idetified and subsequently analysed. Given that, a codified solution version of the system was realise and the NSGA-II results were optimal than a conventional GA and HGA results.

7. Summary and Critical Appraisal

7.1 Summary of Achievements and Original Contributions

The following achievements will assist accelerate the research on redundancy allocation problems:

- Integrated performance –reliability allocation method with the proposed NSGA-II optimisation algorithm can be applied as a realistic decision making tool to accurately estimate optimal topology and reliability of a system. This is proven possible for repairable systems by simply ensuring that an appropriate system configuration and selection of appropriate components combination from the available choice of components is implemented.
- Integrated performance –reliability allocation method, in line with the formulated optimisation problem, can realistically calculate various reliability levels of a multi-level system, alongside, with their associated constraints such as cost, weight and FR.
- Considering the choice of various system designs and the task to appropriately select components for each sub-system, the adoption of this integrated design methodolgy is very handy for identification and selection of optimal system topology from all available choices. Analysis of Figure 5.7 and 5.8 with different topologies are one of the practical topology results produced through this method. These results can assist reliability engineers during the analysis and identification of various competing system topolgies with the view to ascertain their reliability with respect to system total cost.
- Apart from the new formulation method, the flexibility of the written codes can be easily validated or configured to solve problems involving any typical real-life system with multi-level and complex arrangement.

- Also during search period, the code functions does not converge prematurly in any search space region, it rather produces a feasible solution. Unlike the conventional GA, it further introduces multi-directional search diversity in the solution space and can improve the chances to ractify the shortfall in conventional GA.

  Additionally, with respect to the reproduction operators (Mutation and Crossover), new reproduction operators with special characteristics to combine building blocks at different levels and mutate a solution at various levels were defined.

7.2 Critical Appraisal

The present research concentrated largely to produce quality and acceptable results concerning RAP. However, it has not completely addressed all the possible challenges in this area. Challenges such as:

- To investigate the performance of the model using empirical data derived from the real manufacturers. Especially as data model used in this work was simply analytical in nature. It will in extension, be a beneficial practice to evaluate and estimate values for the parameters of the model, such as the feasibility factor and the maximum component reliability, from real data and through consultation with design engineers of components and systems for which an application is undertaken.
- All though two good case studies were successfully carried out in this research. However, more industrial case studies needs to be delivered considering limited industrial case studies reported to have been carried out in the literature.

7.3 Future work

The concept of using this proposed integrated performance- reliability optimisation approach is not far advanced on many systems. The next step will be to advance this approach in the following systems:

- non-repairable systems

- Bridge network and Parallel –Series systems in order to promote and encourage the use of this method.
- To make the approach more attractive and less difficult by developing a simplified codes version with probably more functionalities for industrial commercialisation. The present research showed a very difficult process requiring in-depth knowledge of system analysis. This may be unattractive for prospective researchers in this area to painstakingly go through, prior to getting optimal solutions. The simpler and effective a system optimisation model is, the better and attractive it becomes.

References:

Acharyulu, S.P.V & Seetharamaiah, P. (2015) ‘ A framework for safety automation of safety-critical systems operations’, *Safety Science,*77(2015), pp.133-142.

Adam, L. & Dorota, L. (2012) ‘ Roulette- Wheel Selection via Stochastic acceptance’, *Physica A,* 391(2012), pp. 2193-2196.

Agarwal, H. Renaud, J.E. Preston, E.L. et al., (2004) ‘Uncertainty quantification using evidence theory in multidisplinary design optimisation’, *Reliability Engineering & System Safety,* 85(2004),pp.281-294.

Ahmadizar, A. & Soltanpanah, H. (2011) ‘Reliability Optimisation of a series System with multiple-choice and budget constraints using an efficient ant colony approach’, *Expert Systems with Application,* 38(2011), pp.3640-3646.

Ahmed, Q. Faisal, K. Ahmed, S. (2014) ‘Improving safety and availability of complex systems using a risk-based failure assessment approach’, *Loss Prevention in the Process Industries,* 32(2014), pp.218-229.

Ahmed, R. & Abul, W. ‘A new heuristic approach for optimal reconfiguration in distribution systems’ *Electric Power Systems Research,* 81 (2011), pp.282–289.

Ahn, S.K. Kim, I.S. Oh, K. M. (2010) ‘Deterministic and risk-informed approaches for safety analysis of advanced reactors: Part I, deterministic approaches’, *Reliability Engineering & System Safety,* 95(2010), pp.451-458.

Akgun, I. Gumusbuga, F. & Tansel, B. (2015) ‘Risk based facility location by using fault tree analysis in disaster management’, *Omega,* 52(2015), pp. 168-179.

Alemzadeh, S. & Dastghaibyfard, G. (2013) ‘Time and Cost trade-off using multi-objective task scheduling in utility grids’, *IEEE.*

Ali, H. & Arezoo, Z.A. (2014) 'Uncertainty analysis of water supply networks using the fuzzy set theory and NSGA-II', *Engineering Applications of Artificial Intelligence,* 32(2014), pp.270-282.

Alf, I.W. & Erik, A. (2009) 'The effect of task order on the maintainability of object-oriented Software', *Information and Software Technology*, (51), 293-305.

Ankur, P. & Gursaran, S. (2013) 'Automated test data generation for branch testing using genetic algorithm: An improved approach using branch ordering, memory and elitism', *The Journal of Systems and Software,* 86(2013), pp. 1191-1208.

Ardakan, M.A. & Hamadani, A.Z. (2014) 'Reliability- redundancy allocation problem with cold-standby redundancy strategy', *Simulation modelling practice and theory,* 42(2014), pp. 107-118.

Artur, J. L. (2013) 'A new exponential-type distribution with constant, decreasing, increasing, upside-down bathtub and bathtub-shaped failure rate function', *Computational Statistics and Data Analysis,* 64(2013), pp. 149-170.

Balling, R.J. (2000) 'Decision-based design', *Journal of engineering valuation and cost analysis,* pp.189-198.

Balling, R.J. (2003) 'The maximum fitness function: Multi-objective city and regional planning', *Proceeding of Evolutionary multi-criterion optimisation,* pp.1-15.

Barabadi, A. Barabady, J. Markeset, T. (2011) 'Maintaining analysis considering time-dependent and time-independent covariates', *Reliability Engineering and System safety,* 96(2011), pp.210-217.

Berdingyazov, A. Camci, F. Sevkli, M. et al., (2009) 'Economic Analysis of Maintenance Policies for a System', *IEEE.*

Bergeron, H. Curado, E.M.F. Gazeau, J.P. Ligia, M.C.S (2013) 'Symmetric Generalized binomial Distributions', *Mathematical Physics,* 123301(2013).

Bertolini, M. Bevilacqua, M. (2006) ‘A combined goal programming-AHP approach to maintenance selection problem. *Reliability Engineering & System Safety,*91(2006), pp. 839-848.

Bistouni, F. Jahanshahi, M. (2014) ‘Analysing the reliability of shuffle-exchange networks using reliability block diagrams’, *Reliability engineering and system safety,* 132(2014), pp.97-106.

Blanchard, B.S. Verma, D. Peterson, E.L. (1995) *Maintainability.* New York: John Wiley and sons.

Bonnie, S.H. & Donald, C.J. (2001) ‘Reliability Centered Maintenance and Risk Assessment’, IEEE.

Boris, P. & Jessica, S. (2011) ‘A deterministic annular crossover genetic algorithm optimisation for the unit commitment problem’, *Expert Systems with applications,* 38(2011), pp.6523-6529.

Boudali, H. Crouzen, P.& Stoelinga, M. (2007) ‘Dynamic Fault Tree analysis using Input/Output Interactive Markov Chains’, *IEEE.*

Bourouni, K. (2013) ‘Availability assessment of a reverse osmosis plant: Comparison between Reliability Block Diagram and Fault Tree Analysis Methods’, *Desalination,* 313 (2013), pp. 66–76.

Branimir, K. Lamine, R. Dragan, T. et al., (2016) ' Investigation into recurring military helicopter landing gear failure’, *Engineering failure analysis*, 63(2016), pp.121-130.

Bris, R. Chatelet, E & Yalaui, F. (2003) ‘New method to minimise the preventive maintenance cost of Series-Parallel systems’, *Reliability Engineering and System Safety,* 82(2003), pp.247-255.

Bueno, V.C. & Carmo, I.M. (2007) ‘Active redundancy allocation for a k-out-of-n: F system of dependent components’, *European Journal of Operational Research,* 176(2007), pp.1041-1051.

Canale, E. Robledo, F. Romero, P. et al., (2014) ‘Monte Carlo methods in diameter-constrained reliability’, *Optical Switching and Networking,* 14(2014), pp.134-148.

Cao, J. Wang, Q. Shen, Y. (2012) ‘Research on Modelling Method of Complex System Mission Reliability Simulation’, Academy of Armored Forces Engineering, *IEEE*.

Cekyay, B. Ozekici, S. (2015) ‘Reliability, MTTF and Steady-state availability analysis of systems with exponential lifetimes’, *applied mathematical modelling*, 39(2015), pp. 284-296.

Center for chemical process safety (CCPS). (2000). *Guidelines for chemical quantitative risk analysis*. 2nd edn. New York: AICE.

Chandrupatla, T.R. (2009) *Quality and reliability in engineering.* New York: Cambridge university press.

Changyong, Y. (2014) ‘Software Safety Testing Based on STPA’, *Procedia Engineering*, 80(2014), pp. 399-406.

Chapman, J.L. Lu Lu. Anderson-Cook, C.M. (2014) ‘Incorporating response variability and estimation uncertainty into Pareto front Optimisation’, *Computer & Industrial Engineering,* 76(2014), pp. 253-267.

Chen, X. Ren, H. & Bil, C. (2014) ‘Fault Tree analysis for composite structural damages’, *Institute of mechanical engineers,* 228(9),pp.1466-1474.

Chen, Y. Liu, Y. Cui, Y. et al., (2015) ‘Failure mechanism dependence and reliability evaluation of non-repairable system’, Reliability Engineering and System Safety, 138(2015), pp. 273-283.

Cheng, W. Shi, H. Yin, X. et al., (2011) 'An elitism based genetic algorithm for streaming pattern discovery in wireless sensor networks', *IEEE Communications Letters,* 15(4).

Cheng-Wu, C. Kelvin, F.L. Chun.Pin, T. et al., (2012) 'Hazard management and risk design by optimal statistical analysis', *Springer Science,* 64(2012), pp.1707-1716.

Cho, S. E. (2013) 'First- order reliability analysis of slope considering multiple failure modes', *Engineering Geology,* 154 (2013), pp. 98-105.

Christos, B. & Malik, M. (2013) 'Deterministic Feature selection for k- means clustering', *IEEE,*59(9).

Constantinou, E. Naskos, A. Kakarontzas, G. Stamelos. (2015) 'Extracting reusable components: Asemi-automated approach for a complex structures', *Information Processing Letters,* 115(2015), pp. 414-417.

Cristiano, P. Chenet, L.A. Tambara, G.M. et al., (2015) 'Exploring design diversity redundancy to improve resilience in mixed-signal systems', *Microelectronic Reliability,* 55(2015), pp.2833-2844.

Crowe, D. Feinberg, A. & Bunis, C. (2000) *Design for reliability.* Florida: CRC Press LLC.

Dao, C.D. Zuo, M.J. Pandey, M. (2014) 'Selective maintenance for multi-state series–parallel systems under economic dependence', *Reliability Engineering & System Safety,* 121(1), pp. 240-249.

David, H. (2015) 'Estimation in step-stress life tests with complementary risks from the exponentiated exponential distribution under time constraint and its applications to UAV data', *Statistical Methodology,* 23(2015), pp. 103-122.

David, W.C. Chatwattanasiri, N. Wattanapongsakorn, N. Konak, A.(2015) 'Dynamic k-out-of-n system reliability with component partnership', *Reliability Engineering and System safety.*

Deshpande, V.S. & Modak, J.P. (2002) 'Application of RCM to a medium scale industry', *Reliability Engineering and System safety,* 77(2002), pp. 31-43.

Dhillon, B.S. (2002) *Engineering Maintenance: A Modern approach*. USA: CRC press Boca Raton.

Dhillon, B.S. (2006) *Maintainability, Maintenance and Reliability for Engineers*. USA: CRC press Taylor and Francis Group.

Dinesh, U.K. Knezevic, J. & Crocker, J. (1999) 'Maintenance free operating period- an alternative measure to MTBF and failure rate for specifying reliability', *Reliability Engineering and System Safety,* 64(1999), pp.127-131.

Ding, L. Wang, H. Kang, k. et al., (2014) 'A novel method for SIL verification based on system degradation using reliability block diagram', *Reliability Engineering and System Safety,* 132(2014), pp. 36-45.

Ding, S. Linlin, L. Ying, Y. et al., (2016) 'Robust fuzzy observer-based fault detection for nonlinear systems with disturbances', *Neurocomputing,* 174(2016), pp.767-772.

Doyen, L. & Gaudoin, O. (2004) 'classes of imperfect repair models based on reduction of failure intensity or virtual age', *Reliability Engineering and System Safety*, 84(2004), pp. 45-56.

Ebeling, C. (2001) *Introduction to reliability and maintainability engineering*. New York:Tata McGraw-Hill company Ltd.

Eisinger, S. & Rakowsky, U.K. (2001) 'Modelling of uncertainties in reliability centered maintenance- a probabilistic approach', *Reliability Engineering and System Safety,* 71(2001), pp. 159-164.

Enrio, Z. (2007) *an introduction to the basics of reliability and risk analysis.* Singapore: World Scientific Publishing Co. Pte. Ltd.

Ericson, C. A. (2005) *Hazard Analysis Techniques for System Safety*. USA: John Wiley and Sons, Inc. Publication.

Erik, O. Franklin, D.J. Holbrook, L. et al., (2012) *Machinery's Handbook*. 29th edn. USA: Industrial press, Inc. New York.

Filippini, R. & Andres, S. (2014) 'A modelling framework for the resilience analysis of networked systems-of-systems based on functional dependencies', *Reliability Engineering and System Safety,* 125(2014), pp.82-91.

Forche, R. (1990) 'Analysis of Reliability Block Diagrams with multiple Blocks per Component', *IEEE Proceedings of Annual Reliability and Maintainability Symposium.*

Gao, B. Guo, L. Lin, M. et al., (2012) 'Corrective maintenance Process Simulation Algorithm Research Based on Interaction Process', *IEEE.*

Garg, H. Rani, M. Sharma, S.P. et al., (2014) 'Intuitionistic fuzzy optimisation technique for solving multi-objective reliability optimisation problems in interval environment', *Expert system with applications,* 41(2014), pp. 3157-3167.

Geoffrion, A.M. (1968) 'Proper Efficiency and the theory of vector maximization', *Mathematical Analysis and applications,* 22(1968),pp. 618-630.

Ghosh, D. Chakraborty, D. (2014) 'A new Pareto set generating method for multi-criteria optimisation problems', *Operations research letters,* 42(2014), pp. 514-521.

Giagkiozis, I. Fleming, P.J. (2015) 'Methods for multi-objective optimisation: An analysis', *Information sciences,* 293(2015),pp. 338-350.

Giuseppe, C. Giacomo, G. Alberto, L. (2010) ' A predictive maintenance policy with imperfect monitoring', *Reliability Engineering and System Safety,*95(2010), pp.989-997.

Guanjun, L. Chenxu, Z. Jing, Q. et al., (2014) 'Testability integrated evaluation method based on testability virtual test data', *Aeronautics,* 27 (1), pp.85-92.

Haifeng, G. & Asgarpoor, S. (2011) 'Parallel Monte Carlo simulation for reliability and cost evaluation of equipment and systems', *Electric power Systems Research,* 81(2011), pp.347-356.

Hairong, S. & Jame, J.H. (2002) 'The Failure of MTTF in Availability Evaluation', Proceedings Annual Reliability and Maintainability Symposium.

Hanea, D.H. Jagtman, H.M. Van Alphen, L.L.M.M. et al., (2010) 'Quantitative and qualitative analysis of the expert and non-expert opinion in fire risk in buildings', *Reliability Engineering & System Safety,*95(2010),pp.729-741.

Hany, S.M.A. Nicholas, B. Peter, Y. et al., (2009) 'Artificial Neural Networks modelling the prednisolone nanoprecipitation in microfluidic reactors' *European Journal of pharmaceutical Science,* 37(2009), pp.514-522.

Hata, A. Arika, K. kusakabe, S. et al., (2015) 'Using Hazard Analysis Method STAMP/STPA in developing model oriented formal specification toward reliable cloud service', *IEEE.*

Helton, J.C. & Davis, F.J. (2003) 'Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems', *Reliability Engineering & System Safety,*81(2003), pp.23-69.

Hering, C. & Stadtmuller, U. (2012) 'Estimating Archimedean copulas in high demision', *Scandinavian Journal of Statistics Theory and Application.*

Hiroyasu, T. Chino, S. Miki, M. (2007) 'Flexibility of design variables to Pareto-Optimal Solutions in Multi Objective Optimisation Problems', *IEEE.*

Houssin, R. & Coulibaly, A. (2014) 'Safety- based availability assessment at design stage', *Computer and Industrial Engineering,* 70(2014), pp.107-115.

Hsieh, C. (2003) ‘Optimal task allocation and hardware redundancy policies in distributed computing systems’, *European Journal of Operational Research,* 147(2003), pp.430-447.

Huang, Z. Ou, C. Lin, B. et al., (2014) ‘Conditional statistical properties of the complex systems having long-duration memory’, *physica A*, 409(2014), pp. 138-145.

Huang, B. Buckley, B. Kechadi, T. (2010) ‘Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunication’, *Expert Systems with Applications,*37(2010), pp. 3638-3648.

Hyo, K. Jae-Sun, K. youngsoo, K. et al., (2005) ‘Risk Assessment of Membrane Type LNG Storage Tanks in Korea-based on Fault Tree Analysis’, *Korean J. Cheng Eng.*, 22(1), pp.1-8(2005).

Injoong, K. (2010) ‘Reliability Object Model Tree (ROM-Tree): A system design-for-reliability method’, *Microelectronics Reliability*, (50), pp. 438-446.

Jones, D.F. Mirrazavi, S.K. Tamiz, M. (2002) ‘Multi-objective meta-heuristics: An over view of the current state-of-the art', *European Journal of operational research*, 137(2002), pp. 1-9.

Kalyanmoy, D. Amrit, P. Sameer, A. et al., (2002) ‘A fast and Elitist Multiobjective Genetic Algorithm: NSGA-II’, *IEEE Transaction on Evolutionary Computation,* 6(2).

Karim, B. (2013) ‘Availability assessment of a reverse osmosis plant: Comparison between Reliability Block Diagram and Fault tree Analysis Methods’, *Desalination,* 313(2013), pp.66-76.

Katsigiannis, Y.A. Georgilakis, P.S. (2008) ‘optimal sizing of small isolated hybrid power systems using tabu search’, 10(5), pp.1241-1245.

Kaufmann, A. Grouchko, D. and Cruon, R. (1977), *Mathematical models for the study of the reliability systems,* Academic Press.

Keren, N. West, H.H. Rogers, W.J. et al., (2003) 'Use of failure rate databases and process safety performance measurements to improve process safety,' *Journal of Hazardous Materials,* 104, pp. 75-93.

Kitayama, S. & Yamazaki, K. (2012) 'Compromise point incorporating trade-off ratio in multi-objective optimisation', *Applied soft computing,* 12(2012), pp. 1959-1964.

Kiureghian, A.D. & Dakessian, T. (1998) 'Multiple design points in first and second-order reliability', *Structural Safety,* 20, pp. 37-49.

Kiureghian, A.D. & Song, J. (2008) 'Multi-scale reliability analysis and updating of complex systems by use of linear programming', *Reliability Engineering and System safety,* 93(2008), pp. 288-297.

Konak, A. Coit, D.W. Smith, A.E. (2006) 'multi-objective optimisation using genetic algorithms: A tutorial', *Reliability Engineering & System Safety,* 91(2006), pp. 992-1007.

Kong, X. Gao, L. Ouyang, H. et al., (2015) 'Solving the redundancy allocation problem with multiple strategy choices using a new simplified particle swarm optimisation', *Reliability Engineering and System Safety,* 144(2015), pp.147-158.

Konya, M. J. (2012) 'Setting reliability Goals and Assessing Reliability Improvement Program Effectiveness', *IEEE PES.*

Koo, S.R. Poong, H.S. Yoo, J. et al., (2005) 'An effective technique for the software requirements analysis of NPP safety-critical systems, based on software inspection, requirements traceability, and formal specification', *Reliability Engineering and Safety* 89(2005), pp.248-260.

Kumar, R. Izui, K. Masataka, Y. et al., (2008) 'Multilevel Redundancy Allocation Optimisation Using Hierarchical Genetic Algorithm', *IEEE Transactions on reliability,* 57(4).

Lapa, C.M.F. Pererira, C.M.N.A. Barros, M.P. (2006) 'A model for preventive maintenance planning by genetic algorithms based in cost and reliability', *Reliability Engineering & System Safety,* 91(2006), pp. 233-240.

Lavasani, S. M. Zendegani, A. Celik, M. (2015) 'An extension to fuzzy faultree analysis (FFTA) application in petrochemical process industry', *Process safety and environmental protection,* 93(2015), pp. 75-88.

Lee, S.J. and Poong, H.S (2014) 'Design of an integrated Operator Support System for Advanced NPP MCR: Issues and Perspectives', *Springer.*

Leung, K.N.F. Zhang, Y.L. Lai, K.K (2011) 'Analysis for two-dissimilar-component cold standby repairable system with repair priority', Doyen, *Reliability Engineering and System safety,* 96(2011), pp.1542-1551.

Leveson, N. (2004) 'A new accident model for engineering safer systems', *Safety Science*, 42(2004) pp. 237-270.

Leveson, N. (2011) *Engineering a Safer World: Systems Thinking Applied to Safety.* Boston: Massachusetts, MIT Press.

Leveson, N.G & Dulac, N. (2005) 'Safety and Risk-Driven Design in Complex Systems of-Systems', *1st NASA/AIAA Space Exploration Conference.*

Levitin, G. (2007) 'Block diagram method for analysing multi-state systems with uncovered failure', *Reliability engineering and system safety*, 92(2007), pp. 727-734.

Lewis, E.E. (1996) *Introduction to Reliability Engineering*. 2nd Edn. Canada: John Wiley & Sons.

Liang, L.Y. Chao, W.C. (2008) 'The strategies of tabu search technique for facility layout optimisation', *Automation in construction,* 17(2008), pp. 657-667.

Lin, J.G. (1976) 'Multi-objective problems: Pareto-Optimal Solutions by Method of Proper Equality Constraints', *IEEE Transactions on Automatic Control,* 21(5).

Lin, Z. H, Huang,Y. Fang, C. (2015) 'Non-periodic preventive maintenance with reliability thresholds for complex repairable systems', *Reliability Engineering and System safety,* 136(2015), pp. 145-156.

Liu, B. Xu, Z. Xie, M. et al., (2014) 'A value-based preventive maintenance policy for multi-component with continuously degrading components', *Reliability Engineering and System Safety,* 132(2014), pp. 83-89.

Liu, F. & Huang, Z. (20015) 'System Dynamics Based Simulation Approach on Corrective Maintenance Cost of Aviation Equipments', *Procedia Engineering,* 99(2015), pp.150-155.

Liu, P. Zuo, M.J. & Meng, M. (2003) 'Using neural network function approximation for optimal design of continuous-state Parallel-Series systems', *Computers & Operations Research,* 30(2003), pp. 339-352.

Liu, X. Wang, W. & Ping, R. (2015) 'An integrated production, inventory and preventive maintenance model for a multi-product production system', *Reliability Engineering & System Safety,* 137(2015), 76-86.

Liyang, X. & Zheng, W. (2009) 'Load-strength Interference Failure Rate Model', *International Journal of Quality and Safety Engineering,* 16(3), pp. 249-260.

Lu, L. & Lewis, G. (2008) 'Configuration determination for K-out-of-n partially redundant systems', *Reliability Engineering and System safety,* 93(2008), pp.1594-1604.

Lu Lu. Zhengguo, X. Wenhai, W. et al., (2013) 'A new fault detection method for computer networks', *Reliability Engineering and System Safety,* 114(2013), pp. 45-51.

Maciejewski, H. Caban, D. (2008) 'Estimation of repairable system *availability* within fixed time horizon', *Reliability Engineering and System safety,*

Maheri, A. (2014) ‘Acritical evaluation of deterministic methods in size optimisation of reliable and cost effective standalone hybrid renewable energy systems’, *Reliability Engineering & System Safety,* 130(2014), pp.159-174.

Maheri, A 2013, lecture notes in Modern engineering design *ENO758,* The university of Northumbria Newcastle upon tyne, UK.

Mahmood, S. & Maxim, F. (2015) ‘An Optimal age-based group maintenance policy for multi-unit degrading systems’, *Reliability Engineering and System Safety,*134(2015), pp.230-238.

Maitreya, N. & Adershpal, S. (2007) ‘Probablistic fault diagnosis using adaptive probing’, *Computer Science,* 4875(2007), pp.35-49.

Marco, F. Lana, F. Lata, G. (2015) ‘Selection, tournament and dishonesty’, *Journal of economic behaviour and organisation,*110(2015), pp. 160-175.

Marler, R.T. Arora, J.S. (2004) ‘Survey of multi-objective optimisation methods for engineering’, *Struct Multidisc Optim,* 26(2004), pp. 369-395.

Marler, R.T. Arora, J.S. (2010) ‘the weighted sum method for multi-objective optimisation: new insights’, *Struct Multidisc Optim,* 41(2010), pp. 853-862.

Marseguerra, M. & Zio, E. (2009) ‘Reliability engineering: Old problems and new challenges’, *Reliability Engineering & System Safety,* (94), pp. 125-141.

Maxim, B. (2014) ‘convex optimisation for aggregate production planning’, *International Journal of Production Research,* 4(52), pp.1050-1058.

McCorkle, D. Ashlock, D. Corns, S. et al., (2011) ‘Planned tournament selection’, *Optim Eng,* 12(2011), pp.303-331.

Michael, V. Kwasi, A. & Meredith, J.R. (2000) ‘ An evaluation of maintenance policies for flexible manufacturing systems: A case study of Michael’, *International journal of operation and production management,* 20(4), pp. 409-420.

Michael, S.P. and Jon Shah, N. (2014) ‘Convex optimisation of gradient and shim coil winding patterns’, *Journal of Magnetic Resonance,* 20(2014) pp.36-45.

Ming, H. (2012) ‘ The M-Bayesian Credible Limits of the Reliability Derived from Binomial Distribution’, *Communication and in Statistics-Theory and Methods,* 41(2012), pp. 3814-3830.

Misra, A.K. & Misra, N. (2011) ‘A note on active redundancy allocations in k-out-n systems’, *Statistics and probability letters,* 81(2011), pp. 1518-1523.

Mohamed-Larbi, R. & Daoud, A. (2013) ‘A new Techniques for Generating Minimal Cut-Sets in nontrivial Network’, *AASRI Procedia,*5(2013), pp.67-76.

Mohammad, H.Z. Jafar, H. & Roohallah, A. (2014) ‘Disease Diagnosis with a hybrid method SVR Using NSGA-II, *Neurocomputing,* 136(2014), pp.14-29.

Mohammed, B. Nabil, A. Mark, R. (2014) ‘Convex optimisation for monocular visual odometry trajectory estimation’, *Robotica,* (2014), pp.1-20.

Moura, M. LIns, I.S. Droguett, E.L. et al., (2015) ‘A multi objective genetic algorithm for determing efficient risk-based inspection program’, Reliability engineering and system safety, 133(2015), pp. 253-265.

Naama, B. Bouzeboudja, H. Allali, A. (2013) ‘Application of tabu search and genetic algorithm in minimize losses in power system. Using the b-coefficient method’, *Energy Procedia,* 36(2013), pp. 687-693.

Naderpour, M. & Jie Lu, G.Z. (2015) ‘An abnormal situation modelling method to assist operators in safety-critical systems’, *Reliability Engineering and System Safety,* 133(2015), pp.33-47.

Naess, A. Leira, B.J. Batsevych, O. (2009) ‘System reliability analysis by enhanced monte carlo simulation’, *Structural safety*, 31(2009), pp. 349-355.

Nahas, N. Nourelfath, M. & Ait-Kadi, D. (2007) 'Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series-systems', *Reliability Engineering and System safety,* 92(2007), pp. 211-22.

Naidu, K. Mokhlis, H. Bakar, A.H.A. (2014) 'multiobjective optimisation using weighted sum Artificial Bee Colony algorithm for load frequency control', *Electrical Power and Energy Systems,* 55(2014), pp. 657-667.

Nanda, A.K. Hazra, N.K. (2013) 'Some results on active redundancy at component level versus system level', *Operations research,* 41(2013), pp.241-245.

Nandini, K. Debasis, K. Nair, P. et al., (2010) 'The generalized exponential curve rate with covariates', *Journal of Applied Statistics,*37(10), pp.1625-1636.

Netjasov, F. & Janic, M. (2008) 'A Review of Research on Risk and Safety Modelling in Civil Aviation', *Journal of Air Transport Management,* 14(4), pp. 213-220.

Nikolaidis, Efstratios, Ghiocel, D.M & Singhal, S (2005). Engineering design reliability Handbook, Florida, CRC Press LLC.

Nima, G. Lin, M. Murthy, M. et al., (2009) 'A review on Reliability Models with Covariates', *Springer,* Proceedings of the 4th World Congress on Engineering Asset Management, Greece.

Niu, G.Yang, B. & Pecht, M. (2010) 'Development of an optimised condition-based maintenance system by data fusion and reliability-centered maintenance', *Reliability Engineering & System Safety,* 95(2010), pp.786-796.

OGP. (2010) *Risk Assessment Data Directory.* Available at: http://www.scribd.com/doc/43436605/OGP-Risk-Assessment-Data-Directory-Report-No-434-Compiled-2010#scribd (Accessed: 3 November 2015).

Okafor, E.G. Sun, Y. (2012) 'Multi-objective optimisation of a series-parallel system using GPSIA', *Reliability Engineering & System Safety,* 103(2012), pp. 61-71.

Ouyang, H. Gao, L. Li, S. et al., (2015) ‘improved novel global harmony search with a new relaxation method for reliability optimisation problems.

Ouyang, M. Liu, H. Ming-Hui, Y. el al., (2010) ‘Stamp-based analysis on the railway accident and accident spreading: Taking the China-Jiaoji railway accident for example’, *Safety Science*, 48(2010), pp.544-555.

Ouzineb, M. Nourelfath, M. & Gendreau, M. (2008) ‘Tabu search for the redundancy allocation problem of homogenous Series-Parallel multi-state systems’, *Reliability Engineering and System safety*, 93(2008), pp.1257-1272.

Painton, L. Campbell, J. (1995) ‘Genetic Algorithms in Optimization of System Reliability’, *IEEE TRANSACTION ON RELIABILITY*, pp. 44. (2).

Paul, A.T. & David, C.T. (2012) *Applied Reliability*. 3rd edn. USA: Taylor and Francis group.

Peng, R. Levitin, G. Xie, M. et al., (2010) ‘Defending simple series and parallel systems with imperfect false targets’ *Reliability Engineering and System Safety*, 95 (2010), pp. 679–688.

Petrov, Z. Pavel, G. Z. Cardoso, M.P.J. et al., (2013) ‘An Aspect-Oriented Approach for Designing Safety-Critical Systems’, IEEE.

Phuc, D. Alexandre, V. Eric, L.Benoit, L. (2015) ‘ A proactive condition-based maintenance strategy with both perfect and imperfect maintenance action’, *Reliability Engineering and System Safety*, 133(2015), pp.22-32.

Piterka, L. Jana, J. & Martin, L. (2014) ‘Fault Tree Analysis of emergency core cooling system and containment spray system of WWER440/V213’, *IEEE*.

Pittiglio, P. Bragatto, P. Site, C.D. (2014) ‘Updated failure rates and risk management in process industries’, *Energy Procedia*, 45(2014), pp. 1364-1371.

Pourdarvish, A. Ramezant, Z. (2013) ‘cold standby redundancy allocation in a multi-level series system by memetic algorithm’, *Reliability, quality and safety engineering,* 20(3).

Rahman, S. (2011) ‘Decomposition Method for Structural Reliability Analysis Revisited’, *Probabilistic Engineering Mechanic,* 26(2011), pp.357-363.

Ramirez-Marquez, J.E. (2007) ‘Optimization of system reliability in the presence of common cause failures,’ *Reliability and System Safety,* 92(2007), pp. 1421-1434.

Ran, T. & Ching-Ming, T. (2012) ‘ System reliability optimisation model for construction projects via system reliability theory’, *Automation in construction,* 22(2012), pp. 340-347.

Ravi, A.R. (2005) ‘Intelligent Search Methods for Nonelinear Goal programming Problems’, *INFOR,* 43(2), pp.76-96.

Rejc, Z.B. & Marko, C. (2014) ‘An extension of Multiple Greek Letter method for common cause failures modelling’, *Journal of loss prevention in the process industries,* 29(2014), pp.144-154.

Reliability Centered Maintenance Guide for Facilities and Collateral Equipment, Available online at: http://www.hq.nasa.gov/office/codej/codejx/Assets/Docs/NASARCMGuide.pdf [Accessed 3rd September 2014].

Rippel, M. Lubkemann, J. Nyhuis, P. et al., (2014) ‘Profiling as a means of implementing volume-oriented changeability in the context of strategic production management’, *Manufacturing Technology,* 63(2014), pp. 445-448.

Robert, N.H. & Vesely, W.E. (1987). *Fault tree Handbook.* USA: Government printing office.

Ron, B. (2005) ‘Introduction to IEC 61508’, *Health and Safety Executive,* 55.

Rouvroye, J.L. & Van den Bliek, E.G. (2002) 'Comparing Safety analysis techniques', *Reliability Engineering and System Safety,*75(200) pp.289-294.

Saber, H.M. & Ravindran, A. (1996) 'A partitioning gradient based (PGB) algorithm for solving nonlinear goal programming problems', *Computer Operation Research,* 23(2), pp. 141-152.

Sadrzadeh, A. (2012) 'A genetic algorithm with the heuristic procedure to solve the multi-line layout Problem', *Computer & Industrial Engineering,* 62(2012), pp. 1055-1064.

Safari, J. (2012) 'Multi-objective reliability optimisation of series-parallel systems with a choice of redundancy strategies, *Reliability Engineering & System Safety,*108(2012), pp. 10-20.

Sahoo, L. Bhunia, A. K. Kapur, P.K. (2012) 'Genetic algorithm based multi-objective reliability optimisation in interval environment', *Computers & industrial engineering,* 62(2012), pp. 152-160.

Sebastian, M. Sanchez, A. Serradell, V. 'Age-dependent reliability model considering effects of maintenance and working conditions', *Reliability Engineering and System safety,* 64(1999), pp. 19-31.

Selvik, J.T. & Aven, T. (2011) 'A framework for reliability and risk centered maintenance', *Reliability Engineering and System safety,* 96(2011), pp.324-33.

Sharma, A. Tyagi, V.V. Chen, C.R. et al., (2009) 'Review on thermal energy storage with phase change materials applications', *Renewable and Sustainable Energy Reviews,* 13(2009), pp.318-345.

Sharvia, S. & Yiannis, P.(2015) 'Integrating model checking with Hip-HOPS in model-based safety analysis', *Reliability Engineering and System Safety,* 135(2015), pp.64-80.

Shen, C. & Dhillon, B.S. (2011) 'Reliability and availability analysis of a robot-safety system', *quality in maintenance engineering,* 17(2), pp. 203-232.

Singla, N. Jain, K. Sharma, S.K. (2012) 'The Beta Generalized Weibull distribution: Properties and applications', *Reliability Engineering and System safety,* 102(2012), pp.5-15.

Smith, D. (2001) Reliability Maintainability and Risk: Practical Method for Engineers. 6th edn. United Kingdom: Macmillan.

Sohn, S. & Poong, H.S. (2006) 'Testing digital safety system software with a testability measure based on a software fault tree', *Reliability Engineering and System safety,* 91(2006), pp. 44-52.

Soltani, R. Sadjadi, S. Tavakkoli-Moghaddam, R. (2013) 'Robust cold standby redundancy allocation for non-repairable series-parallel systems through Min-Max regret formulation and Benders' decomposition method', *Institution of Mechanical Engineers,* 228(3) 254-264.

Song, J & Kang, W.(2009) 'System reliability and sensitivity under statistical dependence by matrix-based system reliability method', *Structural safety,* 31(2009),pp. 148-158.

Song, T. Bai, X. Wang, Q. Xing, L. (2012) 'Operational Availability Modelling and Simulation Evaluation', *Crown.*

Stacy, D.H. James, C.S. Maranzano, C.J.(2013) 'Inequality-Based reliability Estimates for Complex systems', *Naval Research logistics,* 60(2013).

Storey, N. (1996) *Safety-Critical Computer Systems*. London: Addison Wesley Longman.

Sun, J. Li-feng, X. Du, S. et al., (2008) 'Reliability modelling and analysis of serial-parallel hybrid multi -operational manufacturing system considering dimensional quality, tool degradation and system configuration' *Int. J. Production Economics,* 114 (2008), pp. 149–164.

Sun, X. Leung, H. Li, B. et al., (2014) 'Change impact analysis and changeability assessment for a change proposal: An empirical study', *Systems and software*, 96(2014), pp. 51-60.

Taheriyoun, M. & Moradinejad, S. (2015) 'Reliability analysis of a wastewater plant treatment using fault tree analysis and Monte Carlo simulation', *Springer.*

Technical Manual, 5-698-1.(2007) Reliability/Availability of Electrical and Mechanical Systems for Command, Control, Communications, Computer, Intelligence, Surveillance and Reconnaissance Facilities. Available at: http://armypubs.army.mil (Accessed: 19 February 2015).

Todinov, M. (2005) Reliability and Risk Model: setting reliability requirements. England: John Wiley and sons Limited.

Trivedi, K.S. Kim, D. Ghosh, R. (2012) 'System Availability assessment using stochastic models', *Applied Stochastic Models in Business and Industry,* 29, pp.94-109.

Tsai, Y.Wang, K. Tsai, L. (2004) ' A study of availability centered preventive maintenance for multi-component Systems', *Reliability Engineering and System Safety,* 84(2004), pp. 261-270.

Tsai-Ching, L. & Wojtek, K.P. (2010) 'Discovery of Root Causes of System Failures by means of Analysis of Repair Records', *IEEE.*

Tsarouhas, P.H. Varzakas, T.H. Arvanitoyannis, I.S. (2009) 'Reliability and maintainability analysis of strudel production line with experimental data-A case study', *Food engineering,* 91(2009), pp. 250-259.

Underwood, P. & Patrick, W.(2014) 'Systems thinking, the swiss cheese model and accident analysis: A comparative systemic analysis of the Grayrigg train derailment Using the ATSB, AcciMap and STAMP models', *Accident Analysis and Prevention,* 68(2014), pp. 75-94.

Valdes, J.E. & Romulo, I. Z. (2006) ‘on the optimal allocation of two active redundancies in a two-component series system’, *Operations Research Letters,* 34(2006), pp.49-52.

Valian, E. Tavakoli, S. Mohanna, E. et al., (2013) ‘improved cuckoo search for reliability optimisation problems’, *Computer and industrial engineering*, 64(2013), pp.459-468.

Van der weide, J.A.M. Pandey, M.D. (2015) ‘A Stochastic alternating renewal process model for unavailability analysis of standby safety equipment’, *Reliability Engineering and System safety,*139(2015), pp.97-104.

Vayrynen, J. Mattila, J. Vilenius, M. Ali, M. Valkama, P. Siuko, M. Semeraro, L. (2011) ‘Predicting the runtime reliability of ITER Remote Handling maintenance equipment’, *Fusion Engineering and Design*, 84(2011), pp.2012-2015.

Verlinden, S. Deconinck, G. Coupe, B. (2012) ‘Hybrid reliability model for nuclear reactor safety system’, *Reliability engineering and system safety,* 101(2012), pp.35-47.

Vijayalakshmi, K & Radhakrishnan, S. (2005) ‘Dynamic Routing from One to Group of Nodes Using Elitism Based GA- Novel Multi Parameter Approach’, *IEEE.*

Volkanovski, A. Cepin, M. Mavko, B. (2009) ‘Application of the fault tree analysis assessment of power system reliability’, *Reliability Engineering and System safety,* 94(2009), pp.1116-1127.

Wang, C. Xing, L. Levitin, G. (2014) ‘Explicit and implicit methods for probabilistic common-cause failure analysis’, *Reliability Engineering & System Safety,* 131(11), pp. 175-184.

Wang, K.S. Hsu, F.S. Liu, P.P. (2002) ‘Modelling the bathtub shape hazard rate function in terms of reliability’, *Reliability Engineering and System safety,*75(2002), pp. 397-406.

Wang, Z. Tang, K. and Yao, X. (2010) 'Memetic Algorithm for Multi-Level Redundancy Allocation', *IEEE Transactions on Reliability*,59(4).

Wang, W. AlliedSignal, I. Phoenix, D.B. et al., (2000) 'Confidence limits on the Inherent Availability of Equipment', *Proceedings Annual Reliability and Maintainability Symposium, IEEE.*

Wei, X. Liao, H. & Jin, T. (2014) 'Maximizing system availability through joint decision on component redundancy and spares inventory', *European journal of operational research*, 237(2014), pp. 164-176.

William, V. Dugan, J. Railsback, J. et al., (2002) *Fault tree handbook with Aerospace applications*. Available at: http://www.hq.nasa.gov/office/codeq/doctree/fthb.pdf (Accessed: 27 August 2015).

Xie, L. Wang, Z. (2009) 'Load-strength interference failure rate model', *Reliability, quality and safety Engineering*, 3(2009), pp. 249-260.

Xue, X & Wang, Y. (2015) 'Ontology alignment based on instance using NSGA-II', *Journal of information science*, 41(1), pp. 58-70.

Yahyatabar, A. A. & Jahromi, A.E. (2012) 'Developing a new model for availability of a series repairable system with multiple cold-standby subsystems and optimisation using simulated annealing considering redundancy and repair facility allocation', *Int Syst Assur Eng Manag*, 3(4). pp. 310-322.

Yang, C. (2014) 'Software Safety Testing Based on STPA', *Procedia Engineering*, 80(2014) pp. 399-406.

Yanjun, L. & Wei, W. (2011) 'Research on the system of reliability block diagram design and reliability prediction', *IEEE Engineering design and manufacturing information*.

Yeh, W. Bae, C. Huang, C. (2015) 'A new cut-based algorithm for the multi-state flow network reliability problem', *Reliability Engineering and System safety*, 136(2015), pp. 1-7.

Yevkin, O. (2009) ‘Truncation Approach with the Decomposition Method for System Reliability Analysis’, *IEEE.*

Yiannis, P. Walker, M. David, P. et al., (2011) ‘Engineering failure analysis and design optimisation with Hip-HOPS’, *Engineering Failure Analysis,* 18(2011), pp.590-608.

Yong-Sheng, H. Ling-Ling, Y. Jian-lei, L. (2009) ‘Realization of GA-NN blind equalization algorithm’, *IEEE.*

Yun, W. Y. & Kim, J. W. (2004) ‘Multi-level redundancy optimization in series systems’, *Computers & Industrial Engineering*, 46(2004), pp.337–346.

Zahra, M. Farhad, A. Fazel, A. (2015) ‘Optimal coordination of directional overcurrent relays using NSGA-II’, *Electric Power Systems Research,*119(2015), pp. 228-236.

Zai, W. Tianshi, C. Ke, T. et al., (2009) ‘A Multi-objective Approach to Redundancy Allocation Problem in Parallel-Series Systems’, *IEEE.*

Zhang, Z. Polet, P. Vanderhaegen, P. et al., (2004) ‘Artificial neural network for violation analysis’, *Reliability Engineering and System Safety,* 84(2004), pp. 3-18.

Zhang, L. Chang, H. & Xu, R. (2012) ‘Equal-width partitioning Roulette wheel selection in genetic algorithm’, *IEEE.*

Zhang, J. Zhang, K. (2010) ‘Application of Tabu search heuristic algorithms for the purpose of energy saving in optimal load distribution strategy for multiple chiller water units’, *IEE.*

Zhang, M. Kecojevic, V. & Komljenovic, D (2014) ‘Investigation of haul truck-related fatal accident in surface mining using fault tree analysis, *Safety Science,*65(2014), pp. 106-117.

Zhang, W. & Fujimura, S. (2010) ‘Improved vector evaluated genetic algorithm with archive for solving multiobjective PPS Problems’, IEEE.

Zhang, W. & Liu, Y. (2008) 'Multi-objective reactive power and voltage control based on fuzzy optimisation strategy and fuzzy adaptive particle swarm', *Electrical Power and Energy System,* 30(2008), pp.525-532.

Zhang, Y.L. Guan, J.W. (2009) 'A geometric process repair model for a repairable cold standby with priority in use and repair', *Reliability Engineering and System safety,* 94(2009), pp. 1782-1787.

Zhang, Q. & Li, H. (2007) 'A multi-objective Evolutionary Algorithm Based on Decomposition', *IEE Transactions on Evolutionary Computation,* 11(2007), pp. 6.

Zhao, R. & Liu, B. (2005) 'Standby redundancy optimisation problems with fuzzy lifetimes', *Computers & Industrial Engineering,*49(2005), pp. 318-338.

Zheng, W. & Liu, Y. (2015) ' Architectural Reliability Estimation Using Design Diversity', *IEEE,* 16th International Symposium on Quality Electronic Design.

Zhigang, T. Zuo, M.J. (2006) 'Redundancy Allocation for Multi-State Systems using physical programming and Genetic Algorithms', *Reliability Engineering and System Safety,*91(2006), 1049-1056.

Zhong-Hua, C. Li-Qing, R. Zhi-Yong, L. (2013) ' A RCM Analytical method considering proactive maintenance', *IEEE.*

Zhu, W. Foulandirad, M. Berenguer, C. (2015) 'Condition-based maintenance policies for a combined wear and shock deterioration model with covariates', *Computers and industrial Engineering,* 85(2015), pp.268-283.

Zuolfaghari, H. Hamadani, A.Z. & Ardakan, M.A. (2014) 'Bi-objective redundancy allocation problem for a system with mixed repairable and non-repairable components', *ISA Transaction,* 53(2014), pp. 17-24.