Multi-State System Reliability Design via Particle Evolutionary Swarm Optimization

by

Angel Eduardo Muñoz Zavala

M. Sc., Centro de Investigación en Matemáticas A.C. (2004)

Dissertation submitted to the Computer Science Department
in partial fulfillment of the requirements for the degree of

Doctorado en Ciencias con Orientación en Ciencias de la Computación

at

CENTRO DE INVESTIGACIÓN EN MATEMÁTICAS A.C.,

June 2009

Author Signature.....
Computer Science Department
May 2009

Certified by.....
Dr. Arturo Hernández Aguirre
Computer Science Department, Titular Research Scientist B
Advisor

Certified by.....
Dr. Enrique Raúl Villa Diharce
Probability and Statistics Department, Titular Research Scientist A
Advisor

Authorized by.....
Dr. Arturo Hernández Aguirre
Departmental Coordinator

.

Multi-State System Reliability Design via Particle Evolutionary Swarm Optimization

by

Angel Eduardo Muñoz Zavala

Dissertation submitted to the Computer Science Department
in May 2009, in partial fulfillment of the requirements for the degree of
Doctorado en Ciencias con Orientación en Ciencias de la Computación

.

Abstract

Many real-world systems are composed of multi-state items that have different performance levels and several failure modes. Items are characterized by their cost, performance and reliability. In a multi-state system, the system and/or its items can posses more than two performance levels. Thereby, the system can have several levels of performance from perfect operation to total failure. The system reliability is modeled to find the optimal items configuration that minimizes the cost of the system subject to minimum system performance levels.

In many new designs, item reliabilities are often uncertain. Thus, the associated system reliability are uncertain as well. In this work, the reliability variance of multi-state systems is studied. Applying the Delta method for the propagation of item variances through system structure, a technique for the approximation of variance of the entire system is proposed.

There are a lot of real-world systems that require a special analysis due to their complex structure and importance in everyday life. The vehicle routing problem has been recognized as one of the great success stories of the operations research area. It is faced everyday by thousands of companies and organizations engaged in the delivery and recollection of goods and people. This dissertation studies the vehicle routing problem in presence of unavailable paths, modeling it as a multi-state system.

An evolutionary algorithm, a modified version of the particle swarm optimization algorithm with constraint handling rules, is introduced for solving system reliability optimization problems.

.

Resumen

Muchos sistemas del mundo real (vida diaria) están compuestos de elementos multi-estado que tienen diferentes niveles de desempeño y varios modos de falla. Los elementos son caracterizados por su costo, desempeño y confiabilidad. En un sistema multi-estado, el sistema y/o sus elementos pueden poseer más de dos niveles de desempeño. Por ende, el sistema puede tener varios niveles de desempeño que van del funcionamiento perfecto hasta la falla total. La confiabilidad del sistema es modelada para encontrar la configuración de componentes óptima que minimice el costo del sistema sujeto a niveles de desempeños mínimos.

En muchos diseños nuevos, las confiabilidades de los elementos son a menudo inciertas. Así, la confiabilidad asociada al sistema, también es incierta. En este trabajo se estudia, la varianza de la confiabilidad de sistemas multi-estado. Aplicando el método Delta para la propagación de las varianzas de los elementos a travéss de la estructura del sistema se propone una técnica para la aproximación de la varianza del sistema completo.

Existen muchos sistemas del mundo real que requieren un análisis especial debido a su estructura compleja e importancia en la vida cotidiana. El problema del ruteo de vehículos se ha reconocido como una de las grandes historias de éxito en el área de investigación de operaciones. Éste es enfrentado todos los días por los miles de compañías y organizaciones comprometidos en la entrega y recolección de bienes y personas. Esta disertación estudia el problema del ruteo de vehículos en presencia de caminos indisponibles, modelandolo como un sistema multi-estado.

Un algoritmo evolutivo consistente en una versión modificada del algoritmo de optimización por enjambres de partículas con reglas para el manejo de restricciones, es introducido para resolver los problemas de optimización de la confiabilidad de sistemas.

.

Agradecimientos

La presente esta dedicada a todas aquellas personas que estuvieron en mi entorno a lo largo de mis estudios de posgrado:

En primer lugar quiero agradecer a mi familia, por las bases inculcadas desde pequeño, y con las cuales he llegado hasta aqui.

De la misma forma, agradezco el apoyo brindado por mis amigos de mi ciudad natal, Guadalajara. A pesar de mi ausencia siempre demostraron su amistad y cariño hacia su servidor; lo cual aprecio infinitamente.

Quiero agradecer a mis compañeros, profesores y amigos del CIMAT, que hicieron de mi estancia en el posgrado una experencia memorable; aprendi mucho de todos ustedes.

También quiero dar un agradecimiento especial a mis alumnos, compañeros docentes y amigos de la Universidad de Guanajuato, donde adquiri mucho más que una constancia de experiencia docente; ... obtuve grandes amistades.

Más alla de un agradecimiento, quiero reconocer el esfuerzo y dedicación de mis asesores para formar de mi persona, ... un hombre de ciencia. Mis trabajos, publicaciones, exposiciones y disertación doctoral son un reflejo de sus enseñanzas. Gracias a ustedes estoy preparado para triunfar en mi siguiente etapa, así como lo hemos hecho juntos desde que comence mi maestría.

Quiero agradecer a una persona en especial con quien he compartido grandes momentos que conservaré muy dentro de mi. Gracias por tu apoyo, amistad y amor.

Por último, agradezco el apoyo brindado durante 4 años por CONACYT para el inicio, desarrollo y culminación de esta disertación doctoral bajo el proyecto de beca #215683. Así mismo, agradezco el apoyo brindado durante 2 meses por CIMAT para la finalización de la presente.

Sinceramente... Angel.

.

Contents

List of Figures

.

List of Tables

.

List of Algorithms

.

Chapter 1

Introduction

Real-world systems are becoming more and more complex and sophisticated. Several systems can perform their intended functions at more than two different levels, from perfectly working to completely failed. In such cases, the system failure can lead to decreased ability to perform the goal task, but not necessarily to a complete failure. These kind of systems are known as multi-state systems (MSS)[83].

Considering system attributes, quality has become a mandatory requirement for customer satisfaction and is playing an increasing role in determining the competitiveness of companies. The concept of reliability is relatively new in the engineering disciplines, but it is becoming increasingly important as part of the engineering design process, the establishment of preventive maintenance programs, and others. The common sense perception of reliability is the absence of failures. Therefore, reliability is sometimes referred as quality in the time dimension because is determined by the failures that may or may not occur during the life time of the product.

Reliability plays a very important role for manufacturers and users. A large reliability increases the production costs, affecting the final user. Thereby, the design reliability optimization problems seek to improve reliability at the minimum cost. The redundancy and reliability allocation problem is a classical optimization problem in the area of system reliability. In general, the objective of these problems is to optimize the system design in terms of the number of components and its reliabilities, subject to known constraints on resources as cost, weight, volume, availability, mean time to failure, etc.

Recently, many authors have discussed and analyzed MSS reliability optimization problems [83, 77, 124]. For instance, Lisnianski and Levitin have produced a series of reports on using genetic algorithms (GA) with the universal generating function (UGF) [130] model for the

optimization of MSS [64, 65, 66, 67, 68, 69, 70, 71, 72, 74, 76, 81, 82, 83].

The UGF technique allows one to find the entire multi-state system reliability based on the performance distributions of its items by using algebraic procedures. It has been reported, that the UGF technique works fast in computational optimization procedures [78]. Lisnianski and Levitin systematically present the UGF approach for redundancy optimization for series-parallel multi-state systems, and describe in detail how GA can be used as the optimization engine. They deal with system optimization considering maintenance, such as joint redundancy and maintenance optimization, imperfect preventive maintenance, and optimal replacement scheduling [83].

1.1 Motivation

Levitin and Lisnianski applied the UGF and GA approach to the optimization of multi-state systems with several types of structures, such as bridge topology, sliding window system, linear consecutively connected system, weighted voting systems, systems with two failure modes and networks. A complete overview is given in [83]. However, there are many real-world systems that require special analysis due to their complex structure and importance in daily life. Particularly the vehicle routing problem (VRP), described as the problem of designing least cost routes from one depot to a set of geographically scattered points (cities, stores, warehouses, schools, customers, etc.), occupies a central place in distribution management. VRP is faced every day by thousands of companies and organizations engaged in the delivery and collection of goods and people. It has been recognized as one of the great success stories of operations research and it has been studied widely since the late fifties. In the VRP, the introduction of MSS reliability analysis can ensure the provision of available traffic lines during incidents or unpredictable events, which could degrade the performance of the planned route.

In real-world, the traffic network probabilities will often be approximated based on the availability frequencies of each link. In several applications, system reliabilities are approximated as well, particularly for new and evolving designs. Nevertheless, many methods and models in classical reliability theory assume that all probabilities are known. In probability theory and statistics, the *variance* $Var(X)$ of a sample is a measure of statistical dispersion. Thereby, the *variance* of the estimated system reliability can be used as a measure of the reliability dispersion around its mean value. The reliability variance of binary systems have been studied in some works [13, 51], also for binary systems composed of dependent items [50, 52]. The variance of the system reliability is an unexplored topic in MSS. In this dissertation, a

model without *a priori* life time distribution assumption is proposed based on the UGF model and the Delta method.

An important subset of design optimization problems, as well as the VRP, contains optimization tasks that are difficult to solve with traditional methods because objectives and constraints (in real world problems) may not be analytically treatable. For these reasons, evolutionary algorithms may be an attractive alternative. Evolutionary algorithms have been widely used to solve reliability optimization problems [77, 124, 125]. For instance, GA has been used in many works to solve reliability optimization problems [67, 83, 125]. Nevertheless, no meta-heuristic is versatile, which could always outperform other meta-heuristics in solving all kinds of problems [137]. The problem becomes more complicated in the presence of a considerable number of linear and nonlinear inequality and equality constraints. The evolutionary algorithms are unconstrained search techniques which lack an explicit mechanism to deal with constrained search spaces [89]. Thereby, introducing new constraint-handling evolutionary optimization approaches can benefit practitioners with more options. The particle swarm optimization (PSO) algorithm, proposed by Kennedy and Eberhart in 1995 [55], has triggered a considerable amount of research regarding the development of mechanisms able to incorporate information about the constraints of a problem into the paradigm. This dissertation introduces a new approach based on PSO with constraint-handling for MSS reliability optimization problems.

1.2 Objectives

The following list summarizes the primary objectives of this dissertation:

- **Reliable Vehicle Routing Problem (RVRP)**: This dissertation introduces the MSS reliability theory to the VRP in the presence of unavailable routes (failures) possibly due to car accidents, congestion, holiday parades, flooding, roadworks. As result an alternative route set is obtained and their ocurrence probabilities (reliabilities) are calculated.

- **Variance of MSS reliability**: Based on the UGF, a technique for the estimation of the variance of MSS reliability is developed applying the Delta method for the propagation of the variances of estimated item reliabilities through system structure.

- **Particle Evolutionary Swarm Optimization (PESO)**: The aim is to develop a simple, fast, and efficient algorithm able to solve MSS reliability optimization problems. A

modified version of the PESO algorithm [95] to solve the reliable VRP is presented in this work. Some comparison of PESO against other evolutionary approaches are presented to show the performance of our proposal.

1.3 Contributions

This work contributes to the fields of both evolutionary algorithms and MSS reliability. Succinctly, it can be described as *a development of an evolutionary algorithm and its application to the MSS reliability design problem.* The main contributions of this dissertation are:

- The introduction of MSS reliability analysis to one of the great area of operations research, the VRP. It creates a reliable route able to avoid interrupted traffic flows through alternative links.
- The approximation of the variance of MSS reliability, which can provide more reliable designs and provide important insights to engineering design strategies involving items with uncertain reliability.
- The development of a new bio-inspired algorithm and its application to MSS reliability design problems. The proposal was tested empirically and shown to offer a significant improvement in both performance and robustness.

1.4 Overview

- Chapter 2 introduces the basic concepts of reliabilty and describes the main MSS function structures.
- Chapter 3 presents a detailed review of the universal generating function for modeling MSS.
- Chapter 4 introduces the MSS reliability theory in the analysis of failures into a traffic network of the VRP.
- Chapter 5 provides a brief overview of the approximation of variance in binary systems and describes the approach based on Delta method to approximate the variance of the MSS reliability.

- Chapter 6 detailedly describes the PESO algorithm and its modified vesion to solve RVRP optmization problems.
- Chapter 7 presents artificial MSS design problems to show the peformance of the PESO algorithm against other evolutionary algorithms to solve reliability optimization problems. Besides, the RVRP model and the approximation of MSS reliability variance, is applied in a real-world VRP (Corbi Bakery).
- Chapter 8 details the main contributions of this dissertation and the future work.

.

Chapter 2

Background

The goal of this chapter is to review the theoretical and practical background literature of system reliablity. It will then proceed with a description of binary and multi-state systems.

2.1 Reliability

Reliability is defined in standards like *ISO 8402* and *BS 4778* as *"the ability of an item to perform a requiered function, under given environmental and operational conditions and for a stated period of time"* [47, 6].

Reliability was used for the first time to compare operational safety of airplane engines just after World War I [40]. Reliability is sometimes referred as quality in the time dimension because is determined by the failures that may or may not occur during the life of the product. According to the definition of reliability given above, *failure* can be defined as *"the termination of an item's ability to perform a required function"* [6].

Failure is regarded as a random phenomenon since it occurs at an uncertain time. Therefore, reliability is often reported in terms of a probability. Ebeling [19] defined reliability as *"the probability that a component or system will perform a required function for a given period of time when used under stated operating conditions. It is the probability of a non-failure over time"*.

The following sections introduce several important concepts related with the reliability of binary and multi-state items. This item can be anything from a simple component to a complex system.

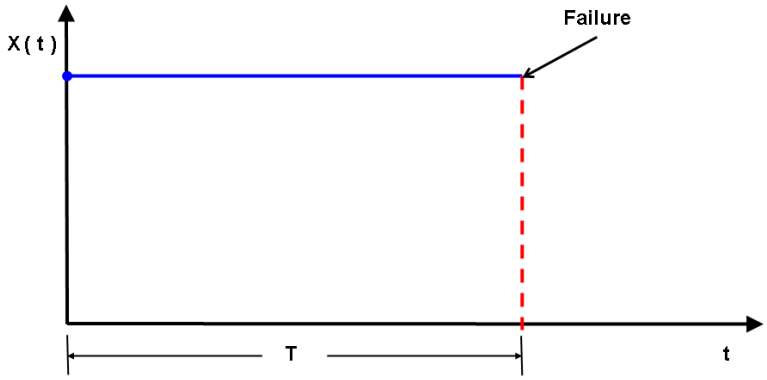


Figure 2.1: Time to failure in binary items

2.2 Binary Items

An item is an entity that is not further subdivided. This imply that an item, in a given reliability study, is regarded as a self-contained unit and is not analyzed in terms of the performance of its constituents.

A binary item possesses two crisp states: perfect functioning and complete failure. Any item is considered in perfect functioning at the starting time $t = 0$. When the item changes from functioning state to failure state, we say that it failed. The item state at time t is expressed by a binary variable $X(t)$, where

$$X(t) = \begin{cases} 1 & functioning \\ 0 & failed \end{cases} \tag{2.1}$$

An important concept emerges here, which referes the time elapsing from when the item is put into operation until it fails for first time, called *time to failure*. It is not necessarily measured in time units. It can be measured by indirect time concepts, such as cycles, distances, counting, etc. The time to failure is modeled as a random variable T because it is subject to chance variations. Figure 2.1 shows the relation between the state variable $X(t)$ and the time to failure T.

Assuming that the time to failure T is continuously distributed with probability density $f(t)$ and distribution function

$$F(t) = P(T \leq t) = \int_0^t f(u)du, t > 0 \tag{2.2}$$

$F(t)$ denotes the probability that the binary item fails within the time interval $(0, t]$. According the definition given by Ebeling [19] (see Section 2.1) reliability is the probability of a non-failure over time

$$R(t) = 1 - F(t) = P(T > t) \tag{2.3}$$

The reliability function $R(t)$ is the probability that the item does not fail in the time interval $(0, t]$ and is also known as the *survivor function*. The reliability can be expressed as the expected value of the state variable $X(t)$

$$E(X(t)) = \sum_{x=0}^{1} xP[X(t) = x] = P[X(t) = 1] = R(t) \tag{2.4}$$

The assumption of binary state variables creates problems for the analysis of units with a gradually decreasing performance. An acceptable limit of performance must be defined and classify all states with a lower level of performance as failure [40]. However, traditional reliability theory fails to represent the true behavior of the item when it has a range of intermediate states that are not accounted for by binary system reliability estimation.

2.3 Multi-state items

A multi-state item can perform their tasks with various distinguished levels of performance or states. The item states can vary as a result of their deterioration, or because of changing ambient conditions, from perfect functioning to complete failure [76].

It is assumed that at the beginning ($t = 0$) the item is in its highest performance (perfect functioning). Failures that lead to decrease the item performance are called partial failures. The item state at time t is expressed by a discrete random variable $G(t)$ that takes its values from the state set

$$g(t) = \{g_0(t), ..., g_n(t)\} \tag{2.5}$$

Generally, $g_0(t)$ represents the complete failure of the item. Whenever the item changes its performance rate, we say that there is a *state transition* in the item. Figure 2.2 shows the relation between the state variable $G(t)$ and the state transition.

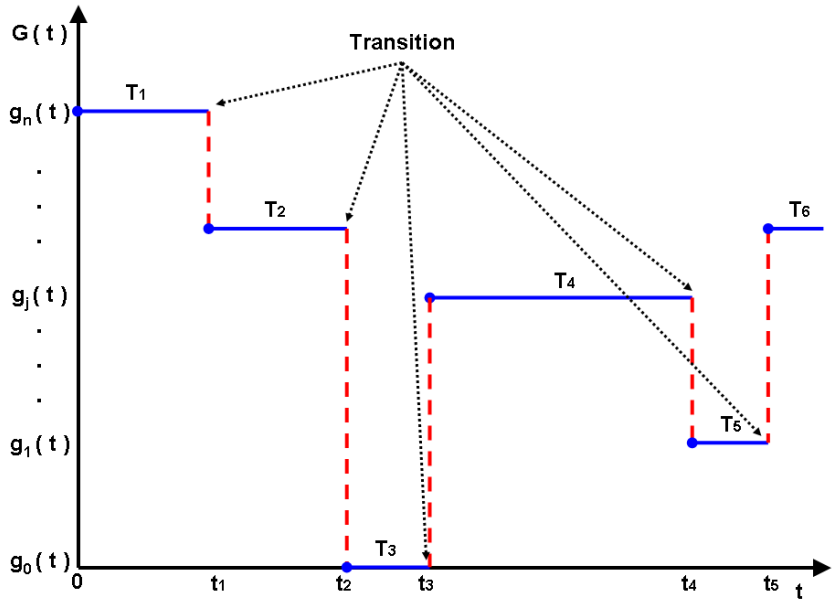


Figure 2.2: State transitions in multi-state items

The probabilities associated with the different states of the item can be represented by the set

$$p(t) = \{p_0(t), ..., p_n(t)\} \tag{2.6}$$

$p_i(t)$ is the probability that the variable state $G(t)$ is in the item state $g_i(t)$ at a specified time t.

$$p_i(t) = P[G(t) = g_i(t)] \tag{2.7}$$

An item can only be in one and only in one of $n + 1$ states. It means that the item states compose the complete group of mutually exclusive events, and then

$$\sum_{i=0}^{n} p_i(t) = 1 \tag{2.8}$$

The collection of pairs $(g_i, p_i), i = \{0, 1, \cdots, n\}$, completely determines the performance probability distribution of the multi-state item.

Item performances are usually associated with the ability of the item to supply a known

demand set

$$d(t) = \{d_0(t), ..., d_m(t)\} \tag{2.9}$$

The probabilities associated with the different demand levels can be represented by the set

$$q(t) = \{q_0(t), ..., q_m(t)\} \tag{2.10}$$

A multi-state item can be associated with two probability functions:

- The item reliability function, $R(t)$, denotes the probability that the item performs its task with a performance rate g_i at a given time t.
- The item availability function, $A(t, w)$, denotes the probability that the item is able to accomplish a demand w at a given time t.

Figure 2.3 presents an example of a item performance and its demand. Note that the item performance is under the demand rate two times. If the demand is not accomplished the item is considered as failed since the item performance should exceed the demand during the stated period of time. Thus, the entire set of possible item states can be divided into two disjoint subsets corresponding to acceptable and unacceptable item functioning.

The relation between the item performance and the demand can be expressed by the acceptability function $F(G, W)$. The acceptable item states correspond to $F(G, W) = 1$ and the unacceptable states correspond to $F(G, W) = 0$. The last equation defines the item failure criterion.

2.4 System Reliability

Real-world systems consist of N components or subsystems (items) and their performance rates are unambiguously determined by the performance rates of these items. System reliability analyze the relation between the items performance of the system and the functioning of the system as a whole. The state of the entire system is determined by the states of its items.

In the binary system reliability analysis, the state of each system item is represented by a binary random variable $X_j(t)$ that indicates the state of item $j \in \{1, 2, \cdots, N\}$ at time t. The states of all N items composing the system are represented by the item state vector

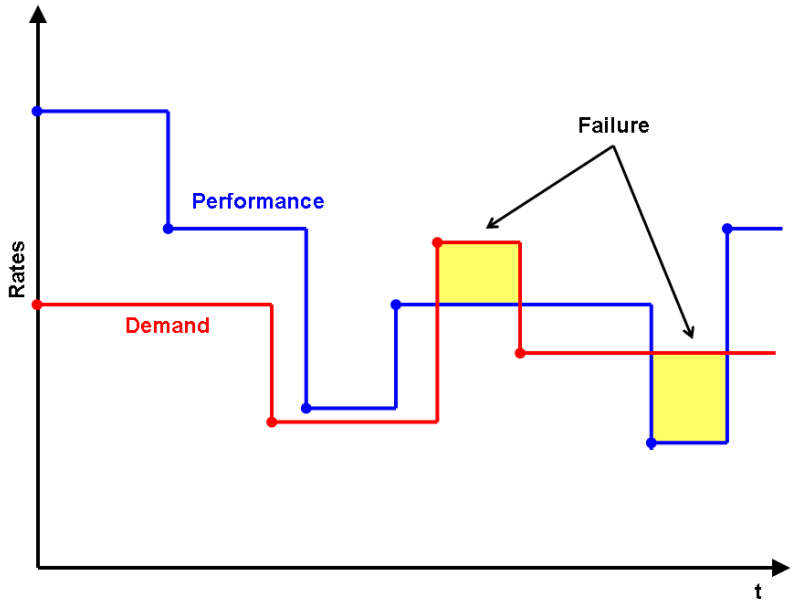


Figure 2.3: Example of item performance and item demand

$< X_1, \cdots, X_N >$. The binary random variable $X(t)$ indicates the state of the entire system at time t.

$$X(t) = \phi(X_1(t), \cdots, X_N(t)) \tag{2.11}$$

The function ϕ is called the system *structure function*. It represents the relation between the item state vector and the system state variable. The reliabilities of the system items compose the item reliability vector $< p_1(t), \cdots, p_N(t) >$. Usually this vector is known, or can be estimated. The system reliability can be calculated based on the system structure function.

The first attempts to build a theory for MSS reliability were made in the late 70s [98, 5]. Proper definitions of a multistate monotone system and of multistate coherent systems were given in the 80s [32]. In 1985, Natvig presents a review of the early development in this area [100]. Recently, the first book on multistate system reliability analysis and optimization was published [83].

In the MSS reliability analysis, the state of each system item is represented by a random variable $G_j(t)$ that indicates the state of item $j \in \{1, 2, \cdots, N\}$ at time t. $G_j(t)$ can represent the performance of a binary or multi-state item. In this respect, MSS can be classified in two categories [114]:

1. MSS that consist of binary items. The MSS performance rate depends on the performance rates of the selected items and their operating state. Every item can process its task with a nominal performance rate: $G_j(t) = g_j$ if it is working, and $G_j(t) = 0$ if it is failed.

2. MSS that consist of multi-state items. The MSS performance rate is dictated by items that have multiple performance rates ranging from their nominal performance rate to the complete failure.

For determining the K MSS performance rates, the states of its N items and its system configuration are used. Assume that $S = \{g_1, ..., g_K\}$ is a space of possible values of the system performance rate. Each system state $k = \{1, 2, \cdots, K\}$ corresponds to a unique combination of the states of the N system items $C^N = \{g_{1,1}, ..., g_{1,n_1}\} \times \{g_{2,1}, ..., g_{2,n_2}\} \times ... \times \{g_{N,1}, ..., g_{N,n_N}\}$. The total number of possible combinations is

$$K = \prod_{j=1}^{N} n_j \tag{2.12}$$

The system performance rate $G(t)$ for each combination $\{G_1(t), ..., G_N(t)\}$ can be found using the system structure function. The system structure function is the transform $\phi(G_1(t), ..., G_N(t)) : C^N \rightarrow M$, which maps the space of items states into the space of system states.

$$G(t) = \phi(G_1(t), ..., G_N(t)) \tag{2.13}$$

Since all of the items are statistically independent, the probability of each unique combination is equal to the product of the probabilities of the components states corresponding to this combination. Therefore, for a system performance rate g_k, corresponding to a combination of N item performance rates $G_j(t)$, its probability p_k can be obtained as

$$p_k(t) = \prod_{j=1}^{N} p_{j,i_j}(t) \tag{2.14}$$

and the system performance in this state k can be obtained as

$$g_k(t) = \phi(g_{1,i_1}(t), ..., g_{N,i_N}(t)) \tag{2.15}$$

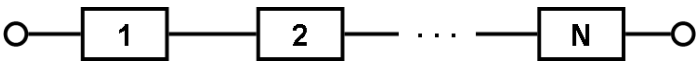


Figure 2.4: Series structure

where i_j is a specific state i for every component j to obtain a system state k.

2.5 Structure Function

The MSS reliability can be obtained by determining the performance distribution of its items and defining the system structure function.

The structure function represents the way the N items are interconnected in a system. It is not necessarily a physical connection. Furthermore, the structure function is concerned with logical interactions between the system items.

It is possible to invent an infinite number of different structure functions for real-world systems. This section presents the structures that are most commonly used in binary and multi-state system reliability analysis [76].

2.5.1 Series Structure

The series connection of system elements represents a case where a total failure of any individual item causes an overall system failure. A series structure of order N is illustrated in Figure 2.4

When an MSS is considered one can distinguish two types of series structures:

- *Transmission:* a system that uses the capacity or productivity of its items as the performance measure. The operation of these systems is associated with flow continuously passing through the items. The item with the minimal transmission capacity becomes the bottleneck of the system as it is shown in Figure 2.5. The bottleneck item determines the system performance $G(t) = min(G_1(t), \cdots, G_N(t))$. In real-world, this kind of systems can be observed mainly in production lines.

- *Processing:* the system performance measure is characterized by an operation time or processing speed. The operation of the systems is associated with consecutive tasks performed by the ordered line of items. The total system operation time is equal to the sum of the operation times of all of its items $G(t) = \sum_{i=1}^{N} G_i(t)$. The complete failure state of a system item corresponds to its processing speed equal to zero, which is equivalent to

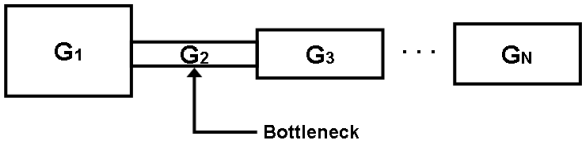


Figure 2.5: Transmission series structure

Figure 2.6: Processing series structure

an infinite operation time. In this case, the operation time of the entire system is also infinite. Real-world systems with processing series structure can be appreciated in service companies (fast food, carwash, package shipping, etc.). Figure 2.6 presents an example of a processing series structure of order N.

2.5.2 Parallel Structure

The parallel connection of system elements represents a case where a system fails if and only if all of its items completely fail. A parallel structure of order N is shown in Figure 2.7

MSS items connected in parallel means that some tasks can be performed by any one of the items. Thus, two basic models of parallel structures are distinguished:

- *Work sharing:* a system that shares the work among its items. The entire system performance rate is equal to the sum of the performance rates of the parallel items $G(t) = \sum_{i=1}^{N} G_i(t)$. Common work sharing systems are the queues in banks, movie theaters, supermarkets, etc. Figure 2.9 illustrates an example of a parallel structure with work sharing of order N.

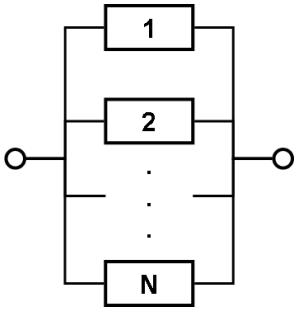


Figure 2.7: Parallel structure

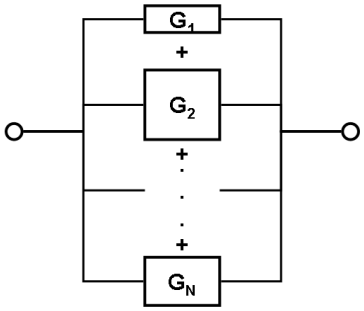


Figure 2.8: Parallel structure with work sharing

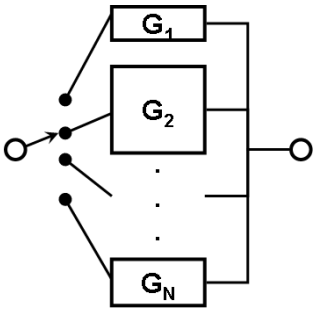


Figure 2.9: Parallel structure without work sharing

- *Without work sharing:*represents a situation where only one item is operating at a time. The system performance rate is equal to the maximal performance rate of the available parallel items $G(t) = max(G_1(t), \cdots, G_N(t))$. For instance, a taxicab radio service sends, from a set of free taxis, the taxi which is nearest to the call. Figure 2.9 presents an example of a parallel structure without work sharing of order N.

2.5.3 K-out-of-N Structure

The K-out-of-N system reliability is defined as the probability that at least K elements out of N are in operable condition. A N-out-of-N system corresponds to the series structure and a 1-out-of-N system corresponds to the parallel structure. In a multi-state generalization of the binary K-out-of-N model, the MSS is in state j if at least K_j items are in state $G_{j,i_j}(t)$ or above [43]. For instance, a car with a V8 engine can walk if at least four cylinders are firing. Figure 2.10 illustrates a logical representation of a 2-out-of-3 system.

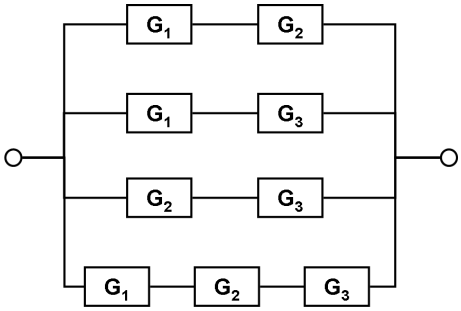


Figure 2.10: 2-out-of-3 structure

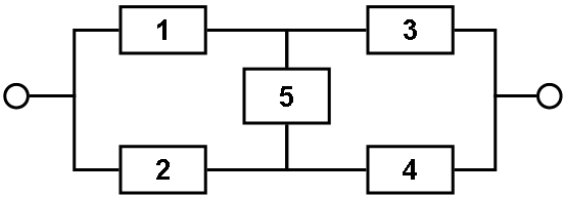


Figure 2.11: Bridge structure

2.5.4 Bridge Structure

The system is designed to ensure the least possible effort both for the functioning and the failed states. As a design strategy, items G_1 and G_2 possess the same operation attributes, but they have different performance levels and probabilities. Similarly, items G_3 and G_4 possess the same operation attributes, and different performance levels and probabilities. Item G_5 works as interface between item pairs (G_1, G_4) and (G_2, G_3). Several security systems have a bridge structure configuration. Figure 2.11 shows a standard bridge structure.

Figure 2.12 illustrates the logical connection of the items to ensure least possible effort for system function.

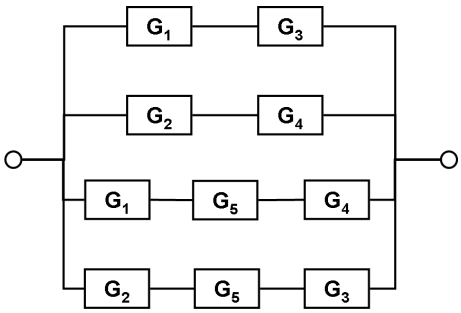


Figure 2.12: Bridge structure: minimum functioning requirements

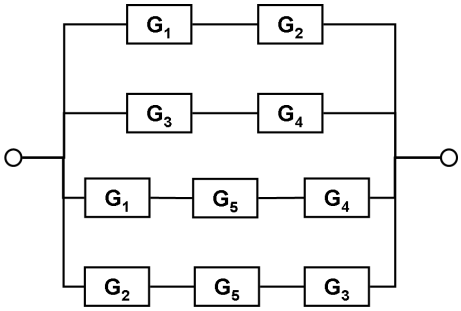


Figure 2.13: Bridge structure: minimum failure requirements

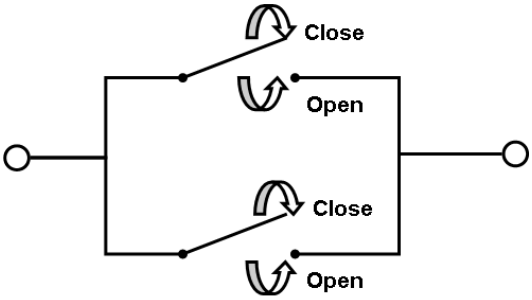


Figure 2.14: System with two failure modes

In Figure 2.13 the logical connection of the items to obtain a completely failed system is shown.

2.5.5 Systems with Two Failure Modes

Systems with two failure modes consist of devices that can fail in either of two different modes. For example, switching systems can fail to close when commanded to close and can fail to open when commanded to open. The system can have different levels of output performance in both modes depending on the states of its items at any given moment. The failures in open and closed modes are mutually exclusive events. Power switches are a typical case of a two failure modes system. An example of a system with two failure modes is presented in Figure 2.14

2.5.6 Weighted Voting Systems

A voting system makes a decision about propositions based on the decisions of N independent individual voting units (items). The voting units can differ in the criteria method applied or the information used. Each unit j ($j = \{1, \cdots, N\}$) produces its decision (output), which can

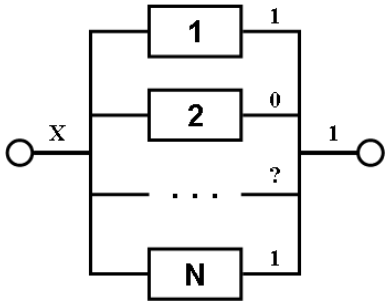


Figure 2.15: Voting system

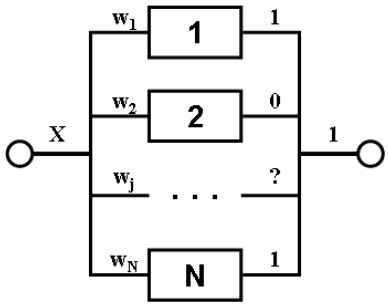


Figure 2.16: Weighted voting system

be $G_j = 1$ (proposition accepted), $G_j = 0$ (proposition rejected), or $G_j =?$ (in the case of abstention).

To make a decision about proposition acceptance, the system incorporates all unit decisions into a unanimous system output which is equal to $G =?$ if all the voting units abstain, equal to $G = 1$ if at least m units produce decision $G_j = 1$, and otherwise equal to $G = 0$. A common voting system applied in medicine is the HIV test which can be composed by the ELISA, Western Blot, and point-of-care tests. Figure 2.15 shows a voting system of order N.

A generalization of the voting system is a weighted voting system where each unit has its own individual weight w_j expressing its relative importance within the system. The system output is $G =?$ if all the units abstain. It is $G = 1$ if the cumulative weight of all $G_j = 1$ units is at least a prespecified fraction τ of the cumulative weight of all non-abstaining units $G_j \neq ?$. Otherwise the system output is $G = 0$. Figure 2.16 presents a weighted voting system of order N.

A generalization of a weighted voting system is a weighted voting classifier consisting of N units (items) where each one provides individual classification decisions $G_j = c$, $c = \{1, \cdots, C, ?\}$. Each unit obtains information about some features of an object to be classified.

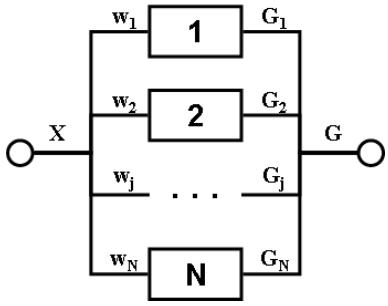


Figure 2.17: Weighted voting classifier

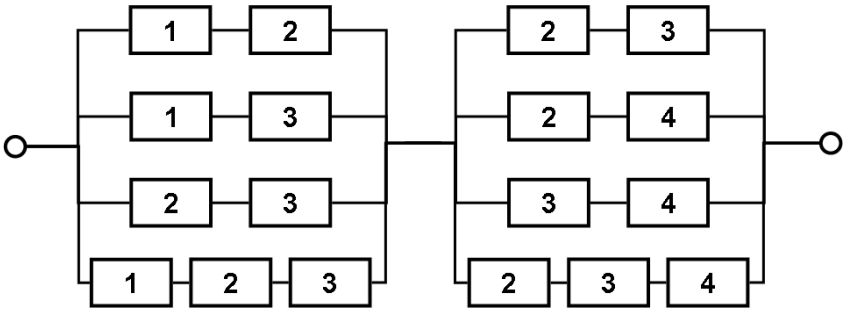


Figure 2.18: Functioning representation of a consecutive 2-out-of-3-from-4 system

Obviously, some units will be highly reliable in recognizing objects of a certain class and much less reliable in recognizing objects of another class.

The entire system output is based on tallying the weighted votes for each decision and choosing the winning one (plurality voting), or the one that has the total weight of supporting votes greater than some specified threshold (threshold voting). The entire system may abstain from voting $G =?$ if no decision ultimately wins. In statistic, different pondered classifier methods are applied to group in several clusters a data set. Figure 2.17 shows a weighted voting system of order N.

2.5.7 Consecutive K-out-of-R-from-N Systems

The binary consecutive K-out-of-R-from-N:F system is composed by N ordered items and fails if at least K out of any R consecutive items fail. There is a consecutive K-out-of-R-from-N:G system that is functioning if at least K out of any R consecutive items operate. A K-out-of-R-from-N:F system is equivalent to an (R-K+1)-out-of-R-from-N:G system. Therefore, we consider only K-out-of-R-from-N:F systems and omit F from their denomination. Figure 2.18 shows the structure representation of a 2-out-of-3-from-4 system.

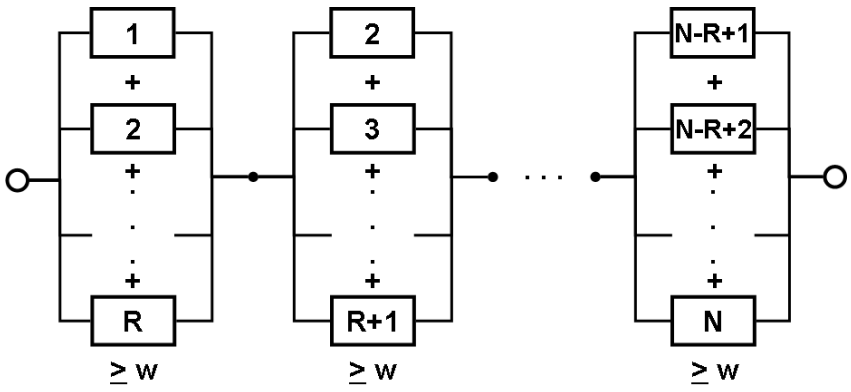


Figure 2.19: Sliding window system

The sliding window system model is a multi-state generalization of the binary consecutive K-out-of-R-from-N system. In this generalized model, the system consists of N linearly ordered multi-state items. The system fails if the sum of the performance rates of any R consecutive items is lower than a defined demand w. For instance, a closed circuit television (CCTV) must ensure the covering of the main areas all time. The logical representation of a sliding window system of order N is illustrated in Figure 2.19.

2.5.8 Consecutive K-out-of-N Systems

The binary consecutive K-out-of-N:F system has N ordered items and fails if at least K consecutive items fail. In the binary consecutive K-out-of-N:G system its N items are connected in such a way that the system works if and only if at least K consecutive items are working. The K-out-of-N:F and K-out-of-N:G systems are duals of each other [60]. The same algorithms for reliability evaluation can be applied to both types of system. Therefore, we consider only K-out-of-N:F systems and omit F from their denomination.

There are two types of consecutive K-out-of-N systems. The linear model corresponds to a consecutive K-out-of-N system as shown in Figure 2.20. The model in which the elements are circularly connected so that the first G_1 and the last G_N items become adjacent to each other is named a circular consecutive K-out-of-N system. It is illustrated in Figure 2.21.

A linear consecutively connected system is a multi-state generalization of the binary linear consecutive K-out-of-N system [45]. In the multi-state model, when an item j is in state $G_j(t) = g_{j,i}(t)$ it is able to provide connection with i following items; $(j+1, j+2, \cdots, j+i)$ items following the one are assumed to be within its range. The linear multi-state consecutively connected system fails if no path exists between its first $j = 1$ and last $j = N$ items. An example

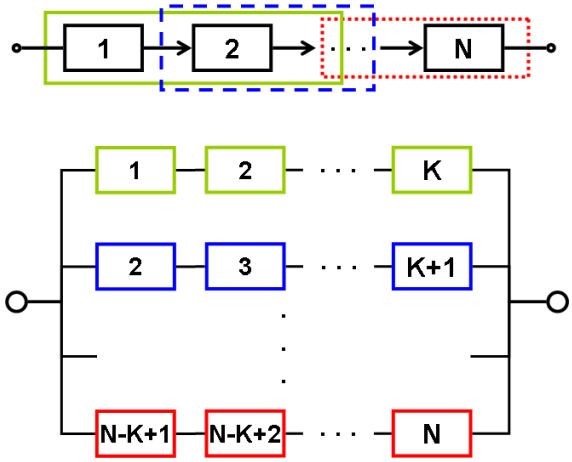


Figure 2.20: Linear consecutive K-out-of-N system

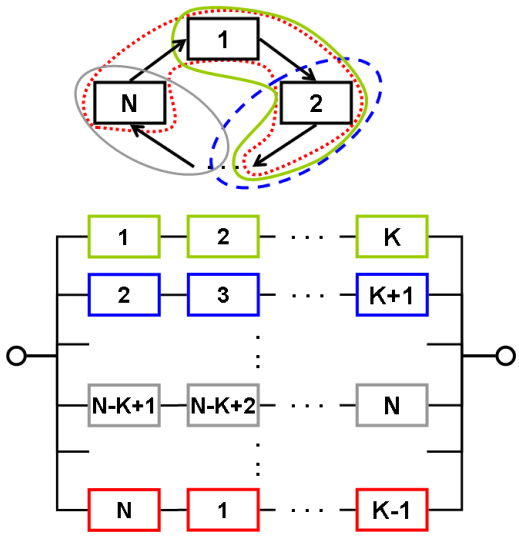


Figure 2.21: Circular consecutive K-out-of-N system

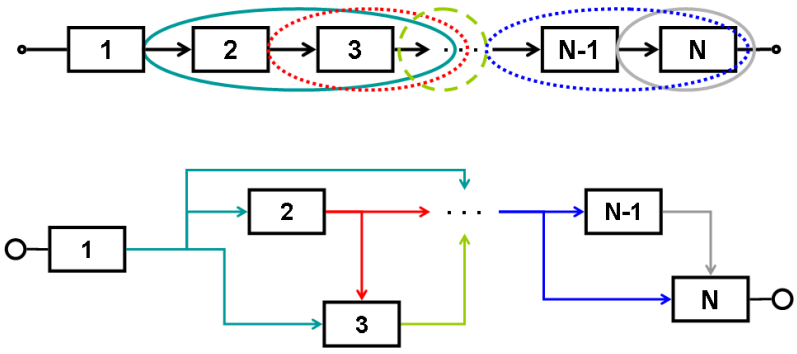


Figure 2.22: Linear consecutively connected system

of a linear consecutively connected system are the TV and radio transmitters composed by a main stations and several relay stations (repeaters or translators). A general example of a linear consecutively connected system is shown in Figure 2.22.

Obviously, there is a multi-state generalization of the binary circular consecutive K-out-of-N system, it is called the circular consecutively connected system. As in the linear consecutively connected system, each item can provide a connection to a different number of items. The system functions if at least one path exists that connects any pair of its items; otherwise there is a system failure.

2.5.9 Networks

Networks are systems consisting of a set of vertices (nodes) and a set of edges that connect these vertices. There are two types of networks: undirected and directed. In the undirected network the edges connect the vertices without any consideration for direction, and in the directed network each edge is followed from one vertex to the next. The networks often have a single root vertex (source) and one or several terminal vertex (sinks). The aim of the networks is the transmission of information or material flow from the source to the sinks. The transmission process intensity depends on the transmission characteristics of the network items and on the availability of these elements. Figure 2.23 illustrates a direct network with one source and two sinks.

The most common network objectives are:

- To maximize the flow between source and sink: the maximal amount of material or information that can be transmitted from the source to the sink through all of the network edges simultaneously.

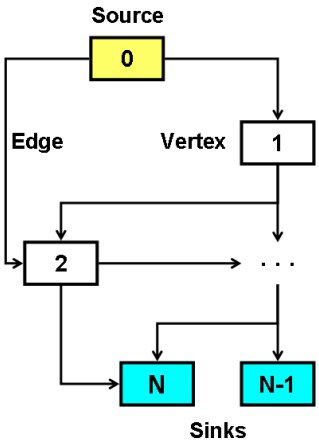


Figure 2.23: Network

- To obtain the single maximal flow path between the source and the sink: the maximal amount of indivisible material or information that can be transmitted through the network by choosing a single path from the source to the sink.
- To minimize the time of transmission between the source and the sink: the delivery delay in networks having edges and/or vertices with limited transmission speed.
- To evaluate the probability that the sinks are connected to the source or the probabilic distribution of the network performance.

According to its stochastic items, in the multi-state case, the networks can be modeled as [86, 87]:

1. Edges model: In the multi-state edges models, the vertices are assumed fully reliable and edges are multi-state items with a given distribution. The edges states can be associated with transmission capacity. For instance, a power supply network composed by an transformer substation (source), electrical lines with different voltage capacities (edges), and energy consumers (vertices).
2. Vertices model: In the multi-state vertices models, the edges are assumed fully reliable and the vertices are multi-state elements. Each vertex state can be associated with a certain delay. For example, a delivery network composed by a depot (source), fixed roadways (edges), and load or unload nodes with stocastic delay times (vertices).
3. Mixed model: In the mixed multi-state models, both the edges and the vertices are multi-state elements. The model is generalized by assuming that the vertices can provide a

connection to a random set of neighbouring vertices and can have random transmission characteristics (capacity or delay) in the edges at the same time. For instance, a watering network composed by a pipe bomb (source), pipes with different capacities (edges), and irrigators with different coverages (vertices).

2.6 System Reliability Assessment

Traditional reliability assumes that a system and its components can be in either a completely working, or a completely failed state only. This condition has facilitated the development of a extensive theory to assess binary system reliability [40].

The MSS reliability assessment are based on different approaches:

- **Boolean models:** In 1980, Caldarola applied the Boolean logic to represent the k states of each item as a binary vector of dependent variables [8]. In 1985, Wood handled the dependence associated with the binary vector introducing a factorization based on constrained Boolean algebra [138]. The binary decision diagram (BDD) introduced by Zang et al. uses a binary fault tree for every system state [142]. The boolean models are extremely time consuming because the number of boolean variables increases drastically with the increase in number of states [83].

- **Minimal cuts and minimal paths:** A minimal path set contents only the needed items to obtain a specific system state. On the contrary, a minimal cut set contents only the needed items to obtain a failure or decrement in the system performance. Natvig [99], and Hudson and Kapur [44], introduced the first minimal cut-path approches in 1982. Recently, Yeh defines a minimal cut using a node set for evaluating network reliability [140]. Although, there are many algorithms to enumerate minimal cut/path sets, all of these involve advanced mathematics [121].

- **Monte-Carlo simulation:** Rocco and Zio extend the cellular automata-based, Monte Carlo sampling methodology for computing network reliability [118]. Recently, Ramirez-Marquez proposed an algorithm based on two major steps that use a probabilistic discovery approach and Monte Carlo simulation [114]. A Monte-Carlo simulation model may be a fairly true representation of the real world, but the main disadvantage of the simulation technique is the time and expense involved in the development and execution of the model [75].

- **Universal Generation Function (UGF):** The UGF method was introduced in 1986 by Ushakov [130]. It involves intuitively simple recursive procedures combined with simplification techniques. Levitin and Lisnianski have produced a series of reports on using UGF [64, 67, 72, 74, 75, 76, 78, 82]. it has proved to be very effective in evaluating the reliability of different types of MSS [83]. The main assumptions when using the UGF technique was that the system items are mutually statistically independent [75]. This assumption is not true in many real-world systems.

- **Heuristics:** Ramirez-Marquez and Coit developed a heuristic of three steps that can yield good design solutions to the capacitated multi-state series-parallel redundancy allocation problem [112]. First, an initial feasible solution is constructed. Then a methodology is applied to try to improve this initial solution. Finally, from the best solution found, a specified number of new solutions are investigated to improve this solution. Kuo et al. proposed a Tabu search for the redundancy allocation problem for a series-parallel system composed of binary items [62]. They apply a M-tuple integer vector to develop tabu search without the need of the universal generating function. Nevertheless, these approaches are limited to specific design reliability problems.

The procedure used in this thesis for MSS reliability assessment is based on the UGF [130]. An detailed analysis of the UGF is presented in the next chapter.

Chapter 3

The Universal Generating Function in MSS Reliability

The universal generating function (UGF) was introduced by Ushakov in 1986 [130]. In mathematics a *generating function* is a formal power series whose coefficients encode information about a sequence a_n that is indexed by the natural numbers. The ordinary generating function of a sequence a_n is

$$G(a_n; x) = \sum_{n=0}^{\infty} a_n x^n \tag{3.1}$$

If a_n is the probability mass function of a discrete random variable, then its ordinary generating function is called a *probability generating function*. Probability generating functions are often employed because they succinctly describe the sequence of probabilities $P(X = i)$. The UGF represents an extension of the widely known *moment generating function* [130].

3.1 Moment Generating Function

In probability theory and statistics, the moment generating function $M(x)$ generates the moments of the probability distribution by successively differentiating itself with respect to x. Let $M_j(x)$ be the moment generating function of the discrete random variable G_j that can take on a finite number $n_j + 1$ of possible values $g_{j,i} = \{g_{j,1}, ..., g_{j,n_j}\}$ with its corresponding probabilities $p_{j,i} = P(G_j = g_{j,i})$. Note that in the above and below notation, we omit variable t since we consider the analysis for a fixed time t.

$$M_j(x) = E(e^{xG_j}) = \sum_{i=0}^{n_j} e^{xg_{j,i}} p_{j,i} \tag{3.2}$$

The moment-generating function of a random variable uniquely determines its p.m.f. This means that a one-to-one correspondence exists between the p.m.f. and the moment-generating function. Note that $M_j(0) = 1$ since

$$M_j(0) = \sum_{i=0}^{n_j} e^0 p_{j,i} = \sum_{i=0}^{n_j} p_{j,i} \tag{3.3}$$

Provided the moment generating function exists in an open interval around $x = 0$, the *kth* moment is given by

$$E(G_j^k) = M^{(k)}(0) = \frac{d^k M}{dt^k}(0) \tag{3.4}$$

There is another important property of moment generating function of special interest for this work. Let $G_1, G_2, ..., G_N$ be a sequence of independent, and not necessarily identically distributed, random variables, and

$$S_N = \sum_{j=1}^{N} a_j G_j \tag{3.5}$$

where the a_j are constants, then the probability density function for S_N is the convolution of the probability density functions of each of the G_j and the moment generating function for S_N is given by

$$M_{S_N}(x) = \prod_{j=1}^{N} M_j(a_j x) \tag{3.6}$$

Therefore, the moment generating function of the sum of the independent random variables is the product of the individual moment generating functions of these variables.

3.2 z-Transform

Related to the moment generating function are a number of other transforms that are common in probability theory. By replacing the function e^x by the variable z in the moment generating function, we obtain another function called the *z-transform* of the discrete random variable G_j.

$$w_j(z) = E(z^{G_j}) = \sum_{i=0}^{n_j} z^{g_{j,i}} p_{j,i} \tag{3.7}$$

The z-transform preserves some basic properties of the moment generating function:

- $$w_j(1) = \sum_{i=0}^{n_j} 1^{g_{j,i}} p_{j,i} = \sum_{i=0}^{n_j} p_{j,i} = 1 \tag{3.8}$$

- The first derivative of $w_j(z)$, at $z = 1$, is equal to the expected value of G_j.

$$E(G_j) = \sum_{i=0}^{n_j} g_{j,i} p_{j,i} \tag{3.9}$$

- The z-transform of the sum of N independent random variables is the product of the individual z-transforms of these variables $G_1, G_2, ..., G_N$.

$$w_{\sum_j^N}(z) = \prod_{j=1}^{N} w_j(z) \tag{3.10}$$

3.3 Universal Generating Function

The polynomial product of the z-transform, corresponding to the variables $G_1, G_2, ..., G_N$, determines the probability mass function (p.m.f.) of the sum of these variables

$$G = \sum_{j=1}^{N} G_j \tag{3.11}$$

In a similar way one can obtain the z-transform representing the p.m.f. of an arbitrary function f (e.g. sum, product, minimize, power) by replacing the polynomial product by a more general *composition operator* $\otimes_f$ over z-transforms of N independent variables [130]. The

composition operator determines the p.m.f., expressed as z-transform polynomial, using simple algebraic operations over individual z-transform polynomials.

$$\otimes_f (G_1, G_2, \cdots, G_N) = \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_N=0}^{n_N} \left(z^{f(g_{1i_1},\ldots,g_{Ni_N})} \prod_{j=1}^{N} p_{ji_j} \right) \tag{3.12}$$

The technique based on the z-transform model and the composition operators $\otimes_f$ is named universal moment generating function (UGF)[130].

In MSS reliability, the UGF $u_j(z)$ of the discrete random variable G_j, represents the probabilities $p_{j,i}$ and performances levels $g_{j,i}$ for every item j as follows:

$$u_j(z) = \sum_{i=0}^{n_j} z^{g_{ji}} p_{ji}(t) \tag{3.13}$$

For modeling the entire performance of the MSS the u-function $U_(z)$ is used to represent the reliability and performance level of each system state.

$$U(z) = \otimes_f (u_1(z), ..., u_N(z)) \tag{3.14}$$

where $\otimes_f$ represents the MSS structure function. The function f into the composition operator $\otimes_f$ expresses the entire performance of a subsystem in terms of the individual performance of their components. The definition of the function f strictly depends on the type of connection between the components in the reliability diagram sense (i.e. on the topology of the subsystem structure).

3.4 Composition Operator $\otimes_f$

The properties of composition operator $\otimes_f$ strictly depend on the properties of the function $f(G_1, \cdots, G_N)$. Since the procedure of the multiplication of the probabilities in this operator is commutative and associative, the entire operator can also have these properties if the function possesses them [76].

Besides, if the function f possesses the recursive property

$$f(G_1, \cdots, G_N) = f(f(G_1, \cdots, G_{N-1}), G_N) \tag{3.15}$$

Then the operator $\otimes_f$ also possesses this property

$$U(z) = \otimes_f(\otimes_f(u_1(z), ..., u_{N-1}(z)), u_N(z)) \tag{3.16}$$

Thereby, one can obtain the u-function $U(z)$ assigning $U_1(z) = u_1(z)$ and applying the operator $\otimes_f$ iteratively

$$\begin{aligned}
U_2(z) &= \otimes_f(U_1(z), u_2(z)) \\
U_3(z) &= \otimes_f(U_2(z), u_3(z)) \\
\vdots &= \vdots \\
U(z) &= U_N(z) = \otimes_f(U_{N-1}(z), u_N(z)) \\
U(z) &= \otimes_f(\otimes_f(\cdots \otimes_f (U_1(z), u_2(z)) \cdots, u_{N-1}(z)), u_N(z))
\end{aligned} \tag{3.17}$$

The function f in composition operators expresses the entire performance of a subsystem in terms of the individual performance of their items. The definition of the function f strictly depends on the type of connection between the system items (system structure).

Several composition operators $\otimes_f$ were determined for important types of MSS by Lisnianski and Levitin [83].

3.4.1 Series-Parallel MSS Example

Figure 3.1 shows a series-parallel MSS composed by 3 items. The operation of these systems is associated with flow continuously passing through the items.

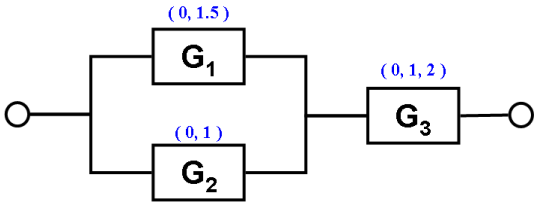


Figure 3.1: Series-parallel example

The vectors shown above the items, in Figure 3.1, represent the states of every item. The u-function $u_i(z)$ for every item can be represented as follows

$$
\begin{aligned}
u_1(z) &= p_{1,0}z^{1.5} + p_{1,1}z^{0.0} \\
u_2(z) &= p_{2,0}z^{1.0} + p_{2,1}z^{0.0} \\
u_3(z) &= p_{3,0}z^{2.0} + p_{3,1}z^{1.0} + p_{3,2}z^{0.0}
\end{aligned}
$$

The items G_1 and G_2 compose a work sharing parallel subsystem. The parallel subsystem and item G_3 posses a transmission series structure. The u-function $U(z)$ represents the entire performance of the series-parallel system shown in Figure 3.1.

$$
U(z) = \otimes_{min}(\otimes_{sum}[u_1(z), u_2(z)], u_3(z))
$$

Applying the compositon operator $\otimes_{sum}$, we obtain

$$
\begin{aligned}
U(z) &= \left[p_{1,0}p_{2,0}z^{1.5+1} + p_{1,0}p_{2,1}z^{1.5+0} + p_{1,1}p_{2,0}z^{0+1} + p_{1,1}p_{2,1}z^{0+0}\right] \overset{\otimes}{}_{min}\, u_3(z) \\
U(z) &= \left[p_{1,0}p_{2,0}z^{2.5} + p_{1,0}p_{2,1}z^{1.5} + p_{1,1}p_{2,0}z^{1} + p_{1,1}p_{2,1}z^{0}\right] \overset{\otimes}{}_{min}\, u_3(z)
\end{aligned}
$$

Finally, we obtain the entire MSS reliability applying the compositon operator $\otimes_{min}$

$$
\begin{aligned}
U(z) &= p_{1,0}p_{2,0}p_{3,0}z^{min(2.5,2.0)} + p_{1,0}p_{2,0}p_{3,1}z^{min(2.5,1.0)} + p_{1,0}p_{2,0}p_{3,2}z^{min(2.5,0.0)} \\
&+ p_{1,0}p_{2,1}p_{3,0}z^{min(1.5,2.0)} + p_{1,0}p_{2,1}p_{3,1}z^{min(1.5,1.0)} + p_{1,0}p_{2,1}p_{3,2}z^{min(1.5,0.0)} \\
&+ p_{1,1}p_{2,0}p_{3,0}z^{min(1.0,2.0)} + p_{1,1}p_{2,0}p_{3,1}z^{min(1.0,1.0)} + p_{1,1}p_{2,0}p_{3,2}z^{min(1.0,0.0)} \\
&+ p_{1,1}p_{2,1}p_{3,0}z^{min(0.0,2.0)} + p_{1,1}p_{2,1}p_{3,1}z^{min(0.0,1.0)} + p_{1,1}p_{2,1}p_{3,2}z^{min(0.0,0.0)} \\
U(z) &= p_{1,0}p_{2,0}p_{3,0}z^{2.0} + p_{1,0}p_{2,0}p_{3,1}z^{1.0} + p_{1,0}p_{2,0}p_{3,2}z^{0.0} + p_{1,0}p_{2,1}p_{3,0}z^{1.5} \\
&+ p_{1,0}p_{2,1}p_{3,1}z^{1.0} + p_{1,0}p_{2,1}p_{3,2}z^{0.0} + p_{1,1}p_{2,0}p_{3,0}z^{1.0} \\
&+ p_{1,1}p_{2,0}p_{3,1}z^{1.0} + p_{1,1}p_{2,0}p_{3,2}z^{0.0} + p_{1,1}p_{2,1}p_{3,0}z^{0.0} \\
&+ p_{1,1}p_{2,1}p_{3,1}z^{0.0} + p_{1,1}p_{2,1}p_{3,2}z^{0.0} \\
U(z) &= \left[(p_{1,1}p_{2,1} + p_{1,1}p_{2,0} + p_{1,0}p_{2,1} + p_{1,0}p_{2,0})p_{3,2} + p_{1,1}p_{2,1}p_{3,1} + p_{1,1}p_{2,1}p_{3,0}\right] z^{0.0} \\
&+ \left[(p_{1,1}p_{2,0} + p_{1,0}p_{2,1} + p_{1,0}p_{2,0})p_{3,1} + p_{1,1}p_{2,0}p_{3,0}\right] z^{1.0}
\end{aligned}
$$

$$+ \quad [p_{1,0}p_{2,1}p_{3,0}]\, z^{1.5} + [p_{1,0}p_{2,0}p_{3,0}]\, z^{2.0} \tag{3.18}$$

3.5 Advantages and Drawbacks

In his book chapter [78], Levitin describes the main advantages of the UGF method:

- **Straightforward.** It is based on intuitively simple recursive procedures and provides a systematic method for the system states enumeration that can replace extremely complicated combinatorial algorithms used for enumerating the possible states in some special types of system (such as consecutive systems or networks).
- **Effective.** Combined with simplification techniques, it allows the system's performance distribution to be obtained in a short time. The UGF technique is fast enough to be implemented in optimization procedures.
- **Universal.** An analyst can use the same recursive procedures for systems with a different physical nature of performance and different types of element interaction.

Besides, the UGF method provides information about other MSS performance indices [78]:

- *Expected performance (W):* The index determines the system's expected performance.

$$W = E[G] = \sum_{k=1}^{K} p_k g_k \tag{3.19}$$

- *Expected acceptability ($R(D)$):* The probability that the MSS satisfies a specific demand. The acceptability of system state can usually be defined by the acceptability function $f(G, D)$ representing the desired relation between the system performance G and system demand D. The acceptability function is $f(G, D) = 1$, if the system performance satisfies the demand, and $f(G, D) = 0$, otherwise.

$$R(D) = E[f(G, D)] = \sum_{k=1}^{K} p_k f(g_k, D) \tag{3.20}$$

- *Conditional expected performance ($W(D)$):* The index determines the system's expected performance given that the system is in acceptable states.

$$W(D) = E[W|f(G,D) = 1] = \sum_{k=1}^{K} p_k g_k \frac{f(g_k, D)}{R(D)} \tag{3.21}$$

Although, the UGF method is powerful computationally efficient tool for the reliability analysis of complex multi-state systems, it possesses the following drawbacks:

- *Independence assumption:* The main assumptions when using the UGF technique was that the system items are mutually statistically independent [75]. This assumption is not true in many real-world systems.
- *Probability mass function (pmf):* The UGF, as a probability-generating function, is a power series representation of the probability mass function of a discrete random variable [136]. A pmf is a function that gives the probability that a discrete random variable is exactly equal to some value. A pmf differs from a probability density function (pdf) in that the values of a pdf, defined only for continuous random variables, are not probabilities as such [54]. Thereby, the UGF can be directly applied to model real-world systems composed of items that possess a stochastic behavior, in terms of a pdf, for the item states.
- *Multi-state node networks (MNN):* The UGF method is known to be very efficient in evaluating Acyclic MNN reliability which is a special MNN without any cycle [73]. However, cycles are very common in networks. There is no proof to show the correctness of the topological numbering of MNN up to now [141]. Yeh proposed a modified UGF method that performs a complete examination of all possible states of each node, which implies its correctness [141].

3.6 UGF in Real-world Systems

The UGF was first applied to MSS reliability by Ushakov in 1988 [131]. The UGF has proven to be very effective for high dimension combinatorial problems [83, 64].

Real-world system reliability analysis generally includes evaluation of complex systems. The explicit derivation of such systems is an extremely complicated task. But, complex systems can be represented as a composition of the structures analyzed in Chapter 2. The reliability block diagram (RBD) method is based on a graphical representation of the system structure.

It recurrent distinguishes subsets of the MSS items and replaces them with single equivalent items.

The u-function of a complex system can be obtained applying recursively the RBD method on its system structure. The recursive approach presumes obtaining u-functions subsystems containing several basic components and then treating every subsystem as a single item using the obtained u-function.

Lisnianski and Levitin pointed out that UGF has two advantages in evaluating MSS reliability [83]:

1. UGF is fast enough to be used in MSS reliability optimization due to its effectiveness in dealing with high-dimension combinatorial problems;
2. UGF calculates the entire MSS performance distribution based on the performance distributions of its items.

The UGF technique allows MSS performance distribution and, thereby, its reliability index to be evaluated based on a fast procedure. The MSS reliability can be obtained as a function of its structure (topology and number of components), performance rates, and reliability values of its components. The UGF technique can be applied for real-world problems of MSS reliability analysis where great number of system states need to be evaluated, which makes it difficult to use traditional techniques [72].

.

Chapter 4

The Reliable Vehicle Routing Problem

A series of applications of MSS reliability theory have been suggested during the last years [103]. For example, it was applied to an offshore electrical power generation system for two nearby oilrigs, where the amounts of power that may possibly be supplied to the two oilrigs are considered as system states [101]. Also, the MSS reliability theory was applied to the Norwegian offshore gas pipeline network in the North Sea, as of the end of the 1980s, transporting gas to Emden in Germany [102]. The system state depends on the amount of gas actually delivered, but also to some extent on the amount of gas compressed mainly by the compressor component closest to Emden. The book *"Multi-state system reliability: assessment, optimization and applications"*, written by Lisnianski and Levitin, contains many examples of application of MSS reliability to real engineering problems [83].

The field of operational research have studied many practical problems related to the delivery of goods and transportation service. A goal of transportation management is to fulfill the requirement of providing goods and/or services from a supply point to various geographically dispersed points with significant economic implications. The vehicle routing problem (VRP) is a well-known combinatorial optimization problem, introduced by Dantzig and Ramser [17], with applications in the field of service operations management, logistic and transportation.

Vehicle Routing Problems (VRP) are all around us in the sense that many consumer products such as soft drinks, beer, bread, snack foods, gasoline, pharmaceuticals, etc., are delivered to retail outlets by a fleet of trucks whose operation fits the vehicle routing model. In practice, the VRP has been recognized as one of the great success stories of operation research,

and has been studied widely since the late fifties.

Solving a VRP means to find the best route to service all customers using a fleet of vehicles. Each customer is to be serviced exactly once by only one vehicle, and each vehicle has a limited capacity. In practice, the problem is aimed at minimizing the total cost of the combined routes for a fleet of vehicles. Since cost is closely associated with distance, the alternative goal is to minimize the distance traveled by a fleet of vehicles with various constraints.

In a real-world environment the travel time on an individual path (between two customers) is stochastic in nature [133]. For instance, car accidents, traffic congestion, holiday parades, flooding, roadworks (maintenance and restoration), cause traffic deviations (route modification). Thereby, alternative links are used to avoid paths temporarily unavaliable. According to environment conditions and locations of unavailable paths, several alternative paths to the optimal route can be obtained, which possess different travel distances or travel costs. This can be represented as a system with a finite number of performance rates.

Many practical systems can perform their intended functions at more than two different levels, ranging from perfectly working to completely failed. That kind of system is known as multi-state system [83]. The performance of such system depends on the performance of its components and the system configuration. The availability of every component for a stated period of time can be used for estimating the system availability.

This work applies the MSS reliability theory to the VRP to account for unavailable paths (failures) in traffic network. As a result a reliable VRP emerges composed by an alternative route set, each one used to avoid an unavaible path along the optimal route.

4.1 The vehicle routing problem

The VRP is a combinatorial optimization problem with applications to distribution management. Every day thousands of companies and organizations are engaged in the delivery and collection of goods and people. The use of computerized methods in distribution processes often result in savings ranging from 5% to 20% in transportation costs [29]. The fact that VRP is both of theoretical and practical interest, explains the amount of attention given to the VRP by researchers in the past years.

The VRP concerns the transport of items between depots and customers by means of a fleet of vehicles. In general, solving a VRP means to find the best route to service all customers using a fleet of vehicles (See Fig. 4.1). The solution must satisfy that all customers are served,

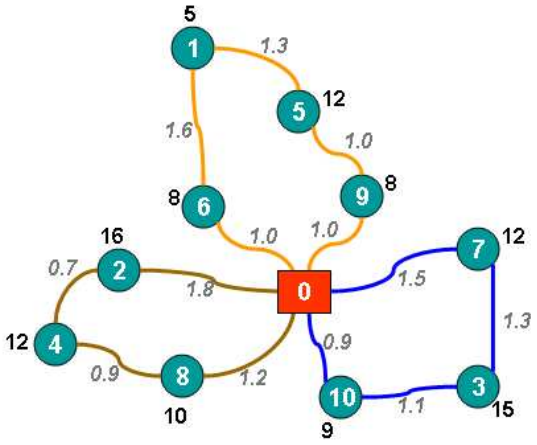


Figure 4.1: VRP example

respecting the operational constraints, and minimizing the total transportation cost.

Figure 4.1 shows a set of scatered points, where point 0 represents the depot and points $1, \ldots, 10$ represent the customers. In Figure 4.1 three routes supply the customer demands. For instance, a route attends customers 2, 4 and 8, whose demands are 16, 12 and 10 units, respectively. This route visits each customer exactly one time and then returns to the starting point (as a Hamiltonian cycle). The arcs that compose the route possess travel costs of 1.8, 0.7, 0.9 and 1.2 units corresponding to the links $0-2$, $2-4$, $4-8$ and $8-0$, respectively. The route acumulates a travel cost (distance, time, expense, etc.) of 4.6 units.

In the VRP, the decisions to be made define the order of the sequence of visits to the customers (route). A route departs from the depot and it is an ordered sequence of visits to be made by a vehicle to the customers, fulfilling their orders. A solution must be verified to be feasible, checking that it does not violate any constraint, such as the one stating that the sum of the demands of the visited vertices shall not exceed the vehicle capacity.

The elements, which define and constrain the model, are: the road network, describing the connectivity among customers and depot; the vehicles, transporting goods between customers and depot on the road network; and the customers, which place orders and receive goods. The VRP is represented by a set of vehicles denoted by V, and a directed graph G, which consist of a set of customers and a depot. The node 0 represents the depot. The set of vertices denoting customers is N. The arc set A denotes all possible connections between the nodes (including the node denoting depot). All routes start and end at node 0. We associate a cost c_{xy} with each arc $(x, y) \in A$ of the routing network. Each vehicle has a capacity limit q_v and each customer n, a demand d_n, $n \in N$. It is common to assume constant speed so that travel distances, travel

times and travel costs are considered as synonymous.

The VRP then calls for the determination of a set of V routes, whose total travel cost is a minimum (Hamiltonian cycles), and such that:

- Each customer is visited exactly once by one cycle.
- Each vehicle starts and ends at the depot.
- The total demand of the customers served by a vehicle is not greater than its capacity.

The complexity of VRP is NP-Hard, and therefore difficult to solve [63]. Due to the nature of the problem it is not viable to use exact approaches for large instances of the VRP. Thus, heuristic and metaheuristic based methods are the only feasible way to provide solutions for industrial scale problems. A recently completed survey of the VRP can be found in [129].

A large number of VRP variants are reported in the literature, depending on the objective function to be optimized and the types of constraints to be satisfied. One of the most important extensions to the VRP is the Vehicle Routing Problem with Time Windows (VRPTW). This variant introduces an additional constraint type: each costumer must be served within a specific time window. Thus, at each node, the service begin-time must be greater than or equal to the beginning of the time window, and the arrival time must be lower than or equal to the end of the time window. When the arrival time is smaller, the vehicle has to wait. Other variants of the problem are multi-depot VRP, VRP with pickup and deliveries, stochastic VRP, etc.

A few works consider real-world enviroment conditions. Most research in this area has focused on routing and scheduling that incorporates variable customer demands. However, there has been very little research on routing and scheduling explicitly incorporating the traffic congestion component into the model. Recently, a dynamic vehicle routing problem with time-dependent travel times due to traffic congestion was presented in [134]. The approach developed introduced the traffic congestion component modeled using a queueing approach to traffic flows. Explicitly making use of the time-dependent congestion results in routes that are shorter in terms of travel time. Fan et al. proposed an adaptive optimal path algorithm based on dynamic programming for the case where optimality is defined as maximum travel time reliability [25]. This work focuses on the risk of encountering undesirable situations and provide a means to capture different travel preferences, specifically late-arrival aversion in the presence of stochastic travel times.

Generally, the VRP solutions understate the interruptions of traffic flows. Most of the models and their solution approaches are far from real-world applications where, for instance,

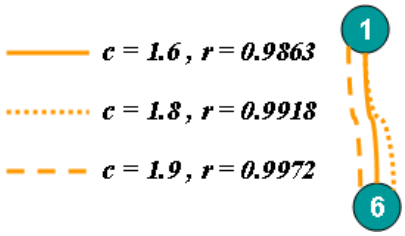


Figure 4.2: Arc subset $a_{6,1}$ in the RVRP

traffic congestion can occur on the used road network. In this paper, the VRP problem considered deals with interrupted traffic flows. The motivation for describing and analyzing congestion for interrupted traffic flows is that the vehicles in the VRP operate in a traffic network which can be interrupted by car accidents, congestion, holiday parades, flooding, roadworks, etc.

Transportation decisions directly affect the total logistic costs. In this context, assigning, scheduling and routing the fleet of a transportation company is a crucial management issue. Compared to the actual realization in real-world, taking into account the interruption of traffic flows results in less time and effort to be invested in replanning in real-time. The approach developed here introduces the multi-state system reliability theory into the standard VRP model.

4.2 MSS Reliability in VRP

For introducing the MSS reliability theory in the VRP, the arc set A, that denotes all possible connections between the nodes in the standar VRP, is extended and divided into arc subsets $a_{xy} \in A$ (items). Every arc subset a_{xy} possesses M_{xy} alternative connections (arcs) between nodes x and y, namely, links l_{xyz}. The arc subsets a_{xy} and a_{yx} can have different sizes. A cost c_{xyz} is associated to each link l_{xyz} of the routing network. For every link $l_{xyz} \in A$, its availabity (reliability) r_{xyz} for a stated period of time is calculated. For instance, a link l_{xyz} unavailable 5 days along a year has a reliability of 360 out of 365 ($r = 0.9863$). In general, for every pair of nodes x-y an arc subset of size M_{xy} is available to avoid interrupted traffic flows. Figure 4.2 presents the arc subset $a_{6,1}$ composed of three links.

Every link $a_{6,1,z}$ shown in Figure 4.2 represents an alternative connection between the nodes 6 and 1. Besides, Figure 4.2 presents the travel cost c and reliability r of every link $l_{6,1,z}$, $z = \{1, 2, 3\}$. In general, each link l_{xyz} represents a multi-state item of the VRP network, which

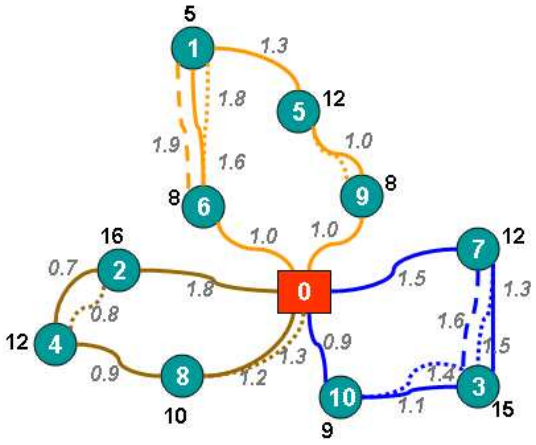


Figure 4.3: RVRP example

is modeled as a MSS in the RVRP approach. The objective is to provide a set of alternative paths to the optimal solution, which can be used as corrective actions when an interruption of traffic flow occurs on the main route. The extra calculation time for large datasets is significant, but is certainly worthwhile if one takes into account the saving delay time in real-time route design. Figure 4.3 shows an example of possible corrective actions in the optimal route presented in Figure 4.1.

The MSS reliability assessment is based on the performance distributions of its components and its structure function. In this paper, the universal generating function (UGF) [130] is used to find the entire MSS performance distribution.

4.3 The Reliable VRP (RVRP)

The aim of this work is to introduce the MSS reliability in the vehicle routing problem. The approach developed here considers the interruption of traffic flow, in a MSS traffic network, modelled through the UGF. The Reliable Vehicle Routing Problem (RVRP) arise whenever some individual paths (i.e. roadways) of the designed route are temporaly unavailables. Common examples are car accidents, roadworks, flooding, and traffic congestions.

The RVRP solution can be modeled as a system with $N+2$ consecutively nodes (customers and depot) C_j, $j \in \{0, 1, \cdots, N+1\}$, where the first node C_0 and the last node C_{N+1} are the depot. At each node C_j, $0 \leq j \leq N$, multi-state items (links) l_{ji}, $i \in \{1, 2, \cdots, M_j\}$ are allocated. These M_j links provide alternative connections between the current node C_j and the

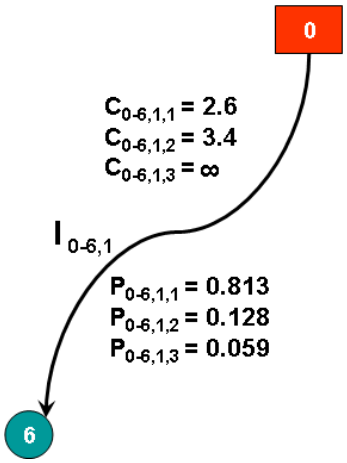


Figure 4.4: Multi-state link example

next node C_{j+1}.

Besides, every link l_{ji} has m_{ji} states, where every state c_{jik} of l_{ji} is a discrete random value with probability mass function:

$$P(l_{ji} = c_{jik}) = p_{jik} \tag{4.1}$$

where $1 \leq k < m_{ji}$, implies that a connection exists with travel cost c_{jik}; and $k = m_{ji}$ implies a total failure state of link l_{ji} with travel cost $c_{jik} = \infty$. Note that different states of link l_{ji} can denote dynamic travel times [134].

The UGF $u_{ji}(z)$ represents all the possible states of the link l_{ji} in node C_j:

$$u_{ji}(z) = \sum_{k=1}^{m_{ji}} p_{jik} z^{c_{jik}} \tag{4.2}$$

The links l_{ji} allocated at node C_j are mutually exclusive events, meaning that the node can always use one and only one of the M_j links. Figure 4.4 shows an example of a multi-state link $l_{0-6,1}$.

It is obvious that the optimal solution of the RVRP include the minimum travel cost between the consecutively nodes C_j and C_{j+1}. Thereby, the UGF $U_j(z)$ that represents the arc subset a_j integrated by the M_j links allocated at node C_j, can be determined as follows:

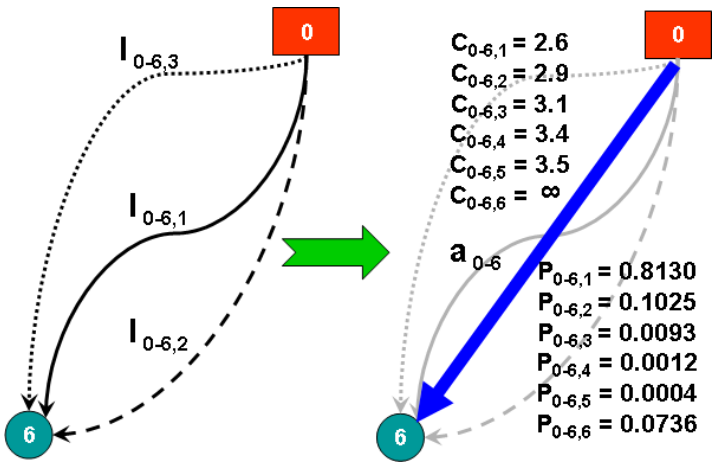


Figure 4.5: Multi-state arc subset example

$$U_j(z) = \otimes_{min}(u_{j1}(z), ..., u_{jM_j}(z)) \tag{4.3}$$

$$U_j(z) = \sum_{k_1=1}^{m_{j1}} \sum_{k_2=1}^{m_{j2}} \cdots \sum_{k_{M_j}=1}^{m_{jM_j}} \left(z^{min(c_{j,1,k_1},...,c_{j,M_j,k_{M_j}})} \prod_{i=1}^{M_j} p_{jik_i} \right) \tag{4.4}$$

Figure 4.5 presents the integration of three multi-state links that compose the arc subset a_{0-6}.

The $N+2$ consecutively ordered nodes of the RVRP can be considered as a MSS with $N+1$ subsystems in serie, since multi-state components are only allocated at nodes $C_j \forall j \leq N$. The total travel cost of the route is equal to the sum of the travel cost at each node C_j. Thereby, the UGF $U(z)$, representing the S alternative routes (states) of the optimal RVRP, can be determined as follows:

$$U(z) = \otimes_{\sum}(U_0(z), ..., U_N(z)) \tag{4.5}$$

$$U(z) = \sum_{h_0=1}^{S_0} \sum_{h_1=1}^{S_1} \cdots \sum_{h_N=1}^{S_N} \left(z^{\sum_{j=0}^{N} c_{jh_j}} \prod_{j=0}^{N} p_{jh_j} \right) \tag{4.6}$$

where $h_j \in \{1, 2, \cdots, S_j\}$ represents the S_j paths (states) of each node C_j. The u-function $U(z)$ provides information about every RVRP state $K \in \{1, 2, \cdots, S\}$: its travel cost $g_K = \sum_{j=0}^{N} c_{jh_j}$ and its reliability (available time) $p_K = \prod_{j=0}^{N} p_{jh_j}$.

The RVRP can be modeled as a standard VRP subject to a overall reliability constraint R. In a typical VRP, the objective is to find a route X with a minimum total travel cost TTC. Therefore, the opimal RVRP is

$$min_X \quad TTC = min(g_1, g_2, \cdots, g_S) \tag{4.7}$$

$$s.t. \quad R \leq 1 - p_{TTC=\infty} \tag{4.8}$$

where X is the route (visitation order), and $p_{TTC=\infty}$ represents the reliability of the total failure state. Similarly, the total reliability R of the route can be determined by the summation of the reliability values at available states $TTC < \infty$.

In brief, the RVRP optimal route represents the solution of a combinatorial problem with an inequality constraint. A RVRP solution consists of S states that represent the optimal route (no traffic interruption) and the alternative routes for every possible interruption traffic event. The alternative routes (states $K = \{2, \cdots, S\}$ obtained by the RVRP, allow to complete the optimal route (state $K = 1$) whenever an interruption occurs in the traffic network, following its determined service sequence for the remaining nodes. Namely, the optimal route $X = \{x_0, x_1, \cdots, x_N\}$ is partially fulfilled from x_0 to x_F $(0 < F < N)$. Then, an alternative path, between nodes x_F and x_{F+1}, is used to complete the optimal route in the same order $x_{F+1}, x_{F+2}, \cdots, x_N$.

In VRP variants, as the VRP with Time Windows (VRPTW), it is neccesary to respect the designed node sequences, because the customers fix a period of time for accomplishing the service. In real-world transportation management, the goods are loaded in the vehicle according to the designed service sequence; thereby, any interruption of the sequence implies an overwork and overtime (unloading and loading) in the route.

Several heuristic and metaheuristic methods have been proposed to solve the standard VRP

[129]. For example, simulated annealing [107], ant colony [117], genetic algorithm [126], etc. For solving the RVRP, the mentioned works or any other approach can be applied introducing the MSS reliability as a constraint. In this dissertation, we use a modified version (discrete) of the Particle Evolutionary Swarm Optimization (PESO) algorithm [94] to solve the RVRP. A detailed description of the PESO algorithm is provided in Chapter 6. For completeness purposes, Chapter 7 presents a real-world RVRP example and its solution with the proposed approach.

Chapter 5

Multi-State System Reliability Variance

The RVRP detailed in Chapter 4 requires information of the state probability for every link between two nodes. In a real-world RVRP, these probabilities will be estimated based on stochastic events like car accidents, floodings, traffic congestions, etc., and deterministic events like holiday parades, roadworks, etc. But, this problem is not exclusive of traffic networks. In general, the probabilities of a real-world system are estimated.

Many methods and models in classical reliability theory assume that all probabilities are known, namely, that every probability involved is perfectly determinable. For many systems, there are limited failure and survival data at the item level. For instance, in the assessment of new system designs, the item reliabilities are not known exactly since must be estimated based on similar components or field-failure reporting database. If the information we have about the functioning of items and systems is based on a statistical analysis, then a probabilistic uncertainty model should be used in order to mathematically represent and manipulate that information. The term variance was first introduced by Fisher to characterize the dispersion among the measures in a given population [26]. Then, the system reliability variance σ^2 must be consider to obtain the best possible bounds for the system reliability given any information about item reliability.

The variance of the system reliability estimate could be regarded as a significant indicator of the estimation uncertainty at the system level. System designers and users generally prefer a system with a high reliability estimate assured by a low uncertainty estimate [88]. But, the reliability optimization methods do not consider uncertainty and can be demonstrated to be

inefficient for some applications. Namely, they lead to solutions that are dominated by other solutions with similar estimated reliability but lower variance. Therefore, maximization of the system reliability and minimization of its estimation uncertainty are mutually important [14].

Reliability practitioners often ignore the estimation variability and treat the system and item reliability as deterministic values. This assumption is misleading if the estimation variability is large. To obtain risk-averse system designs, the variation of the reliability estimate must be explicitly considered. Thus, there is a great need to combine statistical concepts of measuring uncertainty with system reliability modeling techniques to develop truly useful reliability estimation models and optimization algorithms based on real-world limitations on item reliability estimation [49].

5.1 System Reliability Variance

Several works have been proposed to estimate binary system reliability and its variance based on items information. In 1963, Rosenblatt proposed a method based on the asymptotic normality of the maximum likelihood estimate of system reliability, which transforms the system reliability statistic into the standard normal distribution [119]. The variance of the system reliability estimate is estimated based on a Taylor series expansion.

In 1997, Coit proposed an approximate confidence bound for complex system reliability estimate based on log-normal assumption on the system reliability estimate [13]. The approach can be applied to any system that can be decomposed into a series, parallel or series-parallel model. But, the log-normal model is sensitive to the number of subsystems, it yields more accurate estimations when the number of subsystems exceeds eight.

In his dissertation, Jin developed models for system reliability estimations for series-parallel, complex, and network systems [49]. Later, Jin and Coit [50] generalized the log-normal model to systems that allows repeated items within a subsystem while subsystem reliability estimates are statistically independent. This means reliability estimate for duplicated items are dependent.

In 2006, Ramirez-Marquez and Jiang [113] proposed to calculate the variance of series-parallel systems based on the results obtained by Goodman [30]. They descompose a system into a series-parallel structure using the minimal cut/path set technique. The next step is to transform the series or parallel structure into an equivalent two component series or parallel subsystem. This step is applied recursively until reduce the system into two series or parallel configuration. The method is only applicable to systems that can be descomposed into series-

parallel structures.

Recently, Jin and Coit [51] proposed an unbiased version of the model proposed by Coit in 1997 [13] to calculate the variance of a system reliability estimate based on the estimated variance of item reliability estimates.

Few models have been proposed for estimating MSS reliability and its variance. Recently, Ramirez-Marquez and Levitin [115] proposed a model, based on the implementation of the UGF technique, for the estimation of reliability confidence bounds based on item reliability and uncertainty data for MSS with binary-capacitated items.

This chapter presents a method to estimate MSS reliability variance based on the UGF model applying the *Delta method* approximation. The Delta method takes a function that is too complex for analytically computing the variance, creates a linear approximation of that function, and then computes the variance of the simpler linear function that can be used for large sample inference [139].

5.2 The Delta Method

Lets start by considering the formal approach for deriving the variance of a random variable X. In the particular case that the random variable has mean μ, we define the variance as

$$\sigma_X^2 = Var(X) = E[(X-\mu)^2] \tag{5.1}$$

Let Y a new variable defined by $f(X)$. If the probability distribution can be specified, we can use the method of moments to formally derive the mean and variance of the function variable Y. Generally, the variance for functions of random variables are hard to obtain, because they have poorly specified or unspecified distributions. In such cases, we may need other approaches.

The *Delta method* uses the *Taylor series expansion* to approximate the mean and variance of a function of one or more random variables [31]. The Taylor series is a power series expansion of an infinitely differentiable real (or complex) function defined on an open interval around some specified point $x = a$.

$$f(x) \approx f(a) + \frac{f'(a)(x-a)}{1!} + \frac{f''(a)(x-a)^2}{2!} + ... + \frac{f^{(n)}(a)(x-a)^n}{n!} + \cdots \tag{5.2}$$

Thereby, it is possible to approximate the moments of Y, function of a random variable X,

using Taylor expansions; provided that $f(X)$ is sufficiently differentiable and that the moments of X are finite.

The function $f(X)$ can be approximated by a first order Taylor series, where X is evaluated around its mean $X = \mu_X$

$$\hat{f}(X) \approx f(\mu_X) + f'(\mu_X)(X - \mu_X) \tag{5.3}$$

The expected value of Y can be approximated by taking the expected value of $\hat{f}(X)$

$$\begin{aligned} E(Y) &\approx E[\hat{f}(X)] = E[f(\mu_X) + f'(\mu_X)(X - \mu_X)] \\ E(Y) &\approx E[f(\mu_X)] + E[f'(\mu_X)(X - \mu_X)] \\ E(Y) &\approx f(\mu_X) + f'(\mu_X)E[(X - \mu_X)] \\ E(Y) &\approx f(\mu_X) + f'(\mu_X)0 \\ E(Y) &\approx f(\mu_X) \end{aligned} \tag{5.4}$$

The variance of Y can be approximated by taking the variance of $\hat{f}(X)$

$$\begin{aligned} Var(Y) &\approx Var[\hat{f}(X)] = E[\{\hat{f}(X) - E[\hat{f}(X)]\}^2] \\ Var(Y) &\approx E[\{f(\mu_X) + f'(\mu_X)(X - \mu_X) - f(\mu_X)\}^2] \\ Var(Y) &\approx E[f'(\mu_X)^2(X - \mu_X)^2] \\ Var(Y) &\approx [f'(\mu_X)]^2 E[(X - \mu_X)^2] \\ Var(Y) &\approx [f'(\mu_X)]^2 \sigma_X^2 \end{aligned} \tag{5.5}$$

The Delta method is a technique for approximating expected values of functions of random variables when direct evalutaion of the expectation is unfeasible [106]. It assumes that the transformation is approximately linear over the expected range of the parameter.

5.3 Multivariate Delta Method

The Taylor series may also be generalized to functions of more than one variable

$$f(x_1, \cdots, x_n) \approx f(a_1, \cdots, a_n) + \nabla f(a_1, \cdots, a_n)^T [(x_1, \cdots, x_n) - (a_1, \cdots, a_n)] \quad (5.6)$$

The variance of a function Y of N random variables X_i can be approximated by a first order Taylor series

$$\begin{aligned} Var(Y) &\approx \sum_{i=1}^{N} \left[\frac{\partial}{\partial X_i} f(\mu_{X_1}, \cdots, \mu_{X_N}) \right]^2 \sigma_{X_i}^2 \\ &+ 2 \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left[\frac{\partial}{\partial X_i} f(\mu_{X_1}, \cdots, \mu_{X_N}) \right] \left[\frac{\partial}{\partial X_j} f(\mu_{X_1}, \cdots, \mu_{X_N}) \right] Cov(X_i, X_j) \end{aligned} \quad (5.7)$$

Thereby, the Delta method for approximating the variance of a function $f(X_1, \cdots, X_N)$ of independent random variables is

$$Var(\hat{f}(X_1, \cdots, X_N)) = \sum_{i=1}^{N} \left[\frac{\partial}{\partial X_i} f(\mu_{X_1}, \cdots, \mu_{X_N}) \right]^2 \sigma_{X_i}^2 \quad (5.8)$$

5.4 MSS Reliability Variance via Delta-UGF

In Chapter 3, the UGF was used to assess the reliability of any complex MSS. The multi-state items that compose a MSS can be represented by the u-function

$$u_j(z) = \sum_{i=0}^{n_j} z^{g_{j,i}} p_{j,i} \quad (5.9)$$

In this chapter, we assume that the item's probability is estimated $\hat{p}_{j,i}$ and its variance $\sigma_{\hat{p}_{j,i}}^2$ can be known. Thereby, the MSS reliability variance $Var(U(z))$ can be approximated applying the Delta method to the u-function $U(z)$ which represents the entire performance of the MSS.

$$U(z) = \otimes_f (u_1(z), ..., u_N(z)) \quad (5.10)$$

Since the item's states compose the complete group of mutually exclusive events

$$\sum_{i=0}^{n_j} \hat{p}_{j,i} = 1 \quad (5.11)$$

Therefore the probability of any state i can be determined by the n_j remaining probabilities $\hat{p}_{j,s} \forall s \neq i$ [77].

$$\begin{aligned} u_j(z) &= z^{g_{j,0}} \hat{p}_{j,0} + \sum_{i=1}^{n_j} z^{g_{j,i}} \hat{p}_{j,i} \\ u_j(z) &= z^{g_{j,0}} [1 - \sum_{i=1}^{n_j} \hat{p}_{j,i}] + \sum_{i=1}^{n_j} z^{g_{j,i}} \hat{p}_{j,i} \end{aligned} \tag{5.12}$$

This interpretation of the u-function allows to simplify the approximation of the MSS realibility variance. For instance, the u-function of a binary item can be expressed as

$$u_j(z) = \sum_{i=0}^{1} z^{g_{j,i}} \hat{p}_{j,i} = z^{g_{j,0}} \hat{p}_{j,0} + z^{g_{j,1}} (1 - \hat{p}_{j,0}) \tag{5.13}$$

For instance, the variance of the estimated probabilities $\hat{p}_{j,0}$ and $\hat{p}_{j,1}$ of a binary item are $Var(\hat{p}_{j,0}) = \sigma^2_{\hat{p}_{j,0}}$ and $Var(\hat{p}_{j,1}) = Var(1 - \hat{p}_{j,0}) = \sigma^2_{\hat{p}_{j,0}}$.

In general, there is a variance vector σ^2_j for every item $j \in \{1, \cdots, N\}$ that represents the variance of the $\hat{p}_{j,i}$ probabilities of the item.

$$\sigma^2_j = \{\sigma^2_{\hat{p}_{j,1}}, \cdots, \sigma^2_{\hat{p}_{j,n_j}}\} \tag{5.14}$$

The variances of the MSS reliabilities $\hat{r}_k \in \{1, \cdots, K\}$ are associated with the variance vectors σ^2_j, since the MSS u-transform $U(z)$ is function of the item probability vectors $\hat{p}_j$.

$$U(z) = \sum_{k=1}^{K} \hat{r}_k z^{g_k} \tag{5.15}$$

where each reliability $\hat{r}_k$ is expressed as a function of the estimated probabilities $\hat{p}_{j,i}$

$$\hat{r}_k(\hat{p}_{1,1}, \cdots, \hat{p}_{1,n_1}, \cdots, \hat{p}_{N,n_N}) = \sum_{m=1}^{M_k} \prod_{j=1}^{N} p_{j,k_m} \tag{5.16}$$

where $k_m \in \{1, \cdots, n_j\}$ represents the required state of the item j, in the $m-th$ combination, to obtain a performance level (system state) $g_k = f(g_{j,1}, \cdots, g_{j,N})$ (e.g. $f = min$, $f = max$, $f = \sum$, etc.).

Therefore, the MSS reliability variance can be approximated applying the Delta method for every reliability $\hat{r}_k$ to propagate the uncertainty of the probabilities of the N items that composed the system.

$$Var(\hat{r}_k) \approx \sum_{j=1}^{N}\sum_{i=1}^{n_j}\left[\frac{\partial}{\partial \hat{p}_{j,i}}\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}})\right]^2 \sigma^2_{\hat{p}_{j,i}}$$
$$+2\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{l=i+1}^{n_j}\left[\frac{\partial}{\partial \hat{p}_{j,i}}\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}})\right]\left[\frac{\partial}{\partial \hat{p}_{j,l}}\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}})\right] Cov(\hat{p}_{j,i},\hat{p}_{j,l}) \quad (5.17)$$

where $\mu_{\hat{p}_{j,i}}$ is the expectec value or mean of $\hat{p}_{j,i}$; and $Cov(\hat{p}_{j,i},\hat{p}_{j,l})$ is the covariance of the states i and l, $i,l \in \{1,\cdots,n_j\}$, of the item j (mutually exclusive events).

5.4.1 Delta Method: First Order Taylor Expansion

For illustrating the performance of the Delta method, applying a first order Taylor series expansion, the processing series MSS shown in Figure 5.1 is used as example.

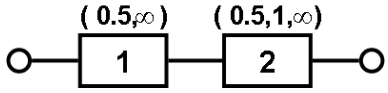


Figure 5.1: Series MSS example

The u-function for each item is

$$u_1(z) = 0.9z^{0.5} + 0.1z^{\infty}$$
$$u_2(z) = 0.7z^{0.5} + 0.2z^{1.0} + 0.1z^{\infty}$$

The u-function $U(z)$ of the entire MSS is

$$\begin{aligned}
U(z) &= \hat{p}_{1,0}\hat{p}_{2,0}z^{1.0} + \hat{p}_{1,0}\hat{p}_{2,1}z^{1.5} + \hat{p}_{1,0}\hat{p}_{2,2}z^{\infty} + \hat{p}_{1,1}\hat{p}_{2,0}z^{\infty} + \hat{p}_{1,1}\hat{p}_{2,1}z^{\infty} + \hat{p}_{1,1}\hat{p}_{2,2}z^{\infty} \\
U(z) &= \hat{p}_{1,0}\hat{p}_{2,0}z^{1.0} + \hat{p}_{1,0}\hat{p}_{2,1}z^{1.5} + \left[\hat{p}_{1,0}\hat{p}_{2,2} + \hat{p}_{1,1}\hat{p}_{2,0} + \hat{p}_{1,1}\hat{p}_{2,1} + \hat{p}_{1,1}\hat{p}_{2,2}\right] z^{\infty} \\
U(z) &= \hat{p}_{1,0}\hat{p}_{2,0}z^{1.0} + \hat{p}_{1,0}\hat{p}_{2,1}z^{1.5} + \left[\hat{p}_{1,0}\hat{p}_{2,2} + \hat{p}_{1,1}(\hat{p}_{2,0} + \hat{p}_{2,1} + \hat{p}_{2,2})\right] z^{\infty}
\end{aligned}$$

$$U(z) = \underbrace{\hat{p}_{1,0}\hat{p}_{2,0}}_{\hat{r}_1} z^{1.0} + \underbrace{\hat{p}_{1,0}\hat{p}_{2,1}}_{\hat{r}_2} z^{1.5} + \underbrace{[\hat{p}_{1,0}\hat{p}_{2,2} + \hat{p}_{1,1}]}_{\hat{r}_3} z^{\infty} \tag{5.18}$$

The variance for every reliability $\hat{r}_k$ can be approximated applying the Delta method for the propagation of the probability variances $\sigma_{\hat{p}_{j,i}}$. First, we model the obtained u-function $U(z)$ of the series system as explained in Eq. (5.12).

$$U(z) = \underbrace{(1-\hat{p}_{1,1})(1-\hat{p}_{2,1}-\hat{p}_{2,2})}_{\hat{r}_1} z^{1.0} + \underbrace{(1-\hat{p}_{1,1})\hat{p}_{2,1}}_{\hat{r}_2} z^{1.5} + \underbrace{[(1-\hat{p}_{1,1})\hat{p}_{2,2} + \hat{p}_{1,1}]}_{\hat{r}_3} z^{\infty} \tag{5.19}$$

Now, applying Eq. (5.17) in Eq. (5.19), we obtain the variance of the reliabilities $\hat{r}_k$.

$$\begin{aligned}
Var(\hat{r}_1) &\approx [-(1-\hat{p}_{2,1}-\hat{p}_{2,2})]^2\sigma^2_{\hat{p}_{1,1}} + [-(1-\hat{p}_{1,1})]^2\sigma^2_{\hat{p}_{2,1}} + [-(1-\hat{p}_{1,1})]^2\sigma^2_{\hat{p}_{2,2}} \\
&+ 2[-(1-\hat{p}_{1,1})][-(1-\hat{p}_{1,1})]Cov(\hat{p}_{2,1},\hat{p}_{2,2}) \\
Var(\hat{r}_2) &\approx [-\hat{p}_{2,1}]^2\sigma^2_{\hat{p}_{1,1}} + [(1-\hat{p}_{1,1})]^2\sigma^2_{\hat{p}_{2,1}} + [0]^2\sigma^2_{\hat{p}_{2,2}} + [(1-\hat{p}_{1,1})][0]Cov(\hat{p}_{2,1},\hat{p}_{2,2}) \\
Var(\hat{r}_3) &\approx [-\hat{p}_{2,2}+1]^2\sigma^2_{\hat{p}_{1,1}} + [0]^2\sigma^2_{\hat{p}_{2,1}} + [(1-\hat{p}_{1,1})]^2\sigma^2_{\hat{p}_{2,2}} \\
&+ [(1-\hat{p}_{1,1})][0]Cov(\hat{p}_{2,1},\hat{p}_{2,2})
\end{aligned} \tag{5.20}$$

Eliminating terms equal to zero ...

$$\begin{aligned}
Var(\hat{r}_1) &\approx [-(1-\hat{p}_{2,1}-\hat{p}_{2,2})]^2\sigma^2_{\hat{p}_{1,1}} + [-(1-\hat{p}_{1,1})]^2\sigma^2_{\hat{p}_{2,1}} + [-(1-\hat{p}_{1,1})]^2\sigma^2_{\hat{p}_{2,2}} \\
&+ 2[-(1-\hat{p}_{1,1})][-(1-\hat{p}_{1,1})]Cov(\hat{p}_{2,1},\hat{p}_{2,2}) \\
Var(\hat{r}_2) &\approx [-\hat{p}_{2,1}]^2\sigma^2_{\hat{p}_{1,1}} + [(1-\hat{p}_{1,1})]^2\sigma^2_{\hat{p}_{2,1}} \\
Var(\hat{r}_3) &\approx [-\hat{p}_{2,2}+1]^2\sigma^2_{\hat{p}_{1,1}} + [(1-\hat{p}_{1,1})]^2\sigma^2_{\hat{p}_{2,2}}
\end{aligned} \tag{5.21}$$

The following variance matrix is assumed for every item:

$$\Sigma_1 = \begin{bmatrix} 0.009 & -0.009 \\ -0.009 & 0.009 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 1.00E-04 & -4.67E-05 & -2.33E-05 \\ -4.67E-05 & 7.62E-05 & -6.67E-06 \\ -2.33E-05 & -6.67E-06 & 4.29E-05 \end{bmatrix} \tag{5.22}$$

Table 5.1: Reliability simulations of the processing series MSS

Entity	State	Peformance	Min	Max	Mean	Variance
Item 1	0	0.5	0.185477	1.000000	0.899914	9.0210E-03
	1	∞	2.7315E-09	0.814523	0.100086	9.0210E-03
Item 2	0	0.5	0.645804	0.743473	0.699999	9.9632e-05
	1	1.0	0.158530	0.247105	0.200002	7.6136e-05
	2	∞	0.072134	0.133480	0.099998	4.2615e-05
MSS	1	1.0	0.127614	0.741661	0.629939	4.5012E-03
	2	1.5	0.036331	0.239675	0.179986	4.2357E-04
	3	∞	0.079080	0.835119	0.190075	7.3429E-03

According to the assumed item probabilities and variance values, the MSS reliability variances are:

$$\begin{aligned} Var(\hat{r}_1) &\approx 4.4957E-03 \\ Var(\hat{r}_2) &\approx 4.2172E-04 \\ Var(\hat{r}_3) &\approx 7.3247E-03 \end{aligned}$$

For validating the Delta method performance, a simulation of the processing series MSS is developed. The *Dirichlet distribution* is used to simulate the probability and variance of every item. Appendix A presents information about the Dirichlet distribution, its probability density function, and its random number generation.

The simulation was performed using the following Dirichlet distributions $Dir(8.1, 0.9)$ and $Dir(1469.3, 419.8, 209.9)$ for every item of the processing series MSS (see Appendix A). The results of $1,000,000$ simulations are shown in Table 5.1.

The MSS reliability variances obtained by the Delta method, accurately approximate the simulation results for the series system. But, the Delta method has a large approximation error for other systems as it is shown below.

5.4.2 The Error in the Delta Method

Figure 5.2 shows a parallel MSS with work sharing.

In the experiment, a variance of $\sigma^2_{\hat{p}_{j,0}} = \sigma^2_{\hat{p}_{j,1}} = 0.009$ for every item is assumed. The u-function for each item is

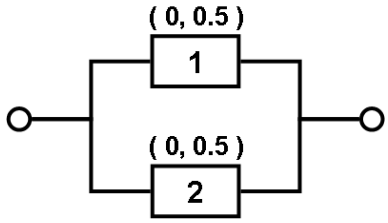


Figure 5.2: Parallel MSS example

$$u_j(z) = 0.9z^{1.0} + 0.1z^{0.0} \tag{5.23}$$

The u-function $U(z)$ of the entire MSS is

$$\begin{aligned} U(z) &= \hat{p}_{1,0}\hat{p}_{2,0}z^{2.0} + \hat{p}_{1,0}\hat{p}_{2,1}z^{1.0} + \hat{p}_{1,1}\hat{p}_{2,0}z^{1.0} + \hat{p}_{1,1}\hat{p}_{2,1}z^{0.0} \\ U(z) &= \hat{p}_{1,0}\hat{p}_{2,0}z^{2.0} + [\hat{p}_{1,0}\hat{p}_{2,1} + \hat{p}_{1,1}\hat{p}_{2,0}]z^{1.0} + \hat{p}_{1,1}\hat{p}_{2,1}z^{0.0} \\ U(z) &= (1-\hat{p}_{1,1})(1-\hat{p}_{2,1})z^{2.0} + [(1-\hat{p}_{1,1})\hat{p}_{2,1} + \hat{p}_{1,1}(1-\hat{p}_{2,1})]z^{1.0} + \hat{p}_{1,1}\hat{p}_{2,1}z^{0.0} \end{aligned} \tag{5.24}$$

Next, the variance of the entire system is approximated applying the Delta method in the MSS reliabilities presented in Eq. (5.24).

$$\begin{aligned} Var(\hat{r}_1) &\approx [-(1-\hat{p}_{2,1})]^2\sigma^2_{\hat{p}_{1,1}} + [(1-\hat{p}_{1,1})]^2\sigma^2_{\hat{p}_{2,1}} \\ Var(\hat{r}_2) &\approx [-\hat{p}_{2,1} + (1-\hat{p}_{2,1})]^2\sigma^2_{\hat{p}_{1,1}} + [(1-\hat{p}_{1,1}) - \hat{p}_{1,1}]^2\sigma^2_{\hat{p}_{2,1}} \\ Var(\hat{r}_3) &\approx [\hat{p}_{2,1}]^2\sigma^2_{\hat{p}_{1,1}} + [\hat{p}_{1,1}]^2\sigma^2_{\hat{p}_{2,1}} \end{aligned} \tag{5.25}$$

The MSS reliability variances are

$$\begin{aligned} Var(\hat{r}_1) &\approx 1.4580E-02 \\ Var(\hat{r}_2) &\approx 1.1520E-02 \\ Var(\hat{r}_3) &\approx 1.8000E-04 \end{aligned}$$

We compare the Delta method performance against the simulation of the parallel MSS using

Table 5.2: Reliability simulations of the parallel MSS with work sharing

Entity	State	Peformance	Min	Max	Mean	Variance
Item 1	0	1.0	0.195248	1.000000	0.899920	8.9874E-03
	1	0.0	1.4005e-10	0.804752	0.100080	8.9874E-03
Item 2	0	1.0	0.165007	1.000000	0.900249	8.9885E-03
	1	0.0	3.6162e-08	0.834993	0.099751	8.9885E-03
MSS	1	2.0	0.146719	0.999968	0.810169	1.4641E-02
	2	1.0	3.2429e-05	0.787680	0.179830	1.1835E-02
	3	0.0	6.9370e-12	0.368007	0.010001	2.5892E-04

the next Dirichlet distribution $Dir(8.1, 0.9)$ for both items of the system (see Appendix A). The results of $1,000,000$ simulations are shown in Table 5.2.

The approximation obtained by Delta method correspond to the simulation results for the states $k = 1$ and $k = 2$ of the parallel MSS. But, the method has an approximation error of 31.12% in the state $k = 3$ with respect to the simulation results.

In general, the Delta method is most accurate when the function $f(X)$ is nearly linear, especially near $E(X)$, and when $Var(X)$ is small. The quality of the estimate of $Var(f(X))$ depends polynomially on $Var(X)$. For small values of σ_X, the Delta method approximation is quite good. For larger values of σ_X, the quality is bad and degrades exponentially.

One immediate option is to use a higher order Taylor series approximation. Including higher order terms in the UGF-Delta method, it can achieve a better approximation to the function variance.

5.4.3 Delta Method: Second Order Taylor Expansion

In the parallel MSS example shown in Figure 5.2, we see that the classical Delta method, which is based on a first order Taylor series expansion of the transformed function $f(X)$, may not do particularly well, if the function is highly non-linear over the range of values being examined, and there is considerable variation in the variables.

One possible solution is to use a higher-order Taylor series approximation; by including higher-order terms, we can achieve a better fit to the function [15].

$$\hat{f}(X) \approx f(\mu_X) + f'(\mu_X)(X - \mu_X) + \frac{1}{2} f''(\mu_X)(X - \mu_X)^2 \tag{5.26}$$

The expected value of $\hat{f}(X)$ is

$$
\begin{aligned}
E(Y) &\approx E\left[f(\mu_X) + f'(\mu_X)(X-\mu_X) + \frac{1}{2}f''(\mu_X)(X-\mu_X)^2\right] \\
E(Y) &\approx f(\mu_X) + \frac{1}{2}f''(\mu_X)\sigma_X^2
\end{aligned} \tag{5.27}
$$

The variance of $\hat{f}(X)$ is

$$
\begin{aligned}
Var(Y) &\approx E[\{\hat{f}(X) - E[\hat{f}(X)]\}^2] \\
Var(Y) &\approx E\left[\{f'(\mu_X)(X-\mu_X) + \frac{1}{2}f''(\mu_X)(X-\mu_X)^2 - \frac{1}{2}f''(\mu_X)\sigma_X^2\right] \\
Var(Y) &\approx [f'(\mu_X)]^2\sigma_X^2 + \frac{1}{4}[f''(\mu_X)]^2 Var(X^2) - [f''(\mu_X)]^2\mu_X^2\sigma_X^2
\end{aligned} \tag{5.28}
$$

The second order Taylor series generalized to a $\hat{r}_k$ function of N independent items composed of n_j mutually exclusive states, can be compactly written as:

$$
\begin{aligned}
&\hat{r}_k(\hat{p}_{1,1},\cdots,\hat{p}_{N,n_N}) \approx \hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}}) + \nabla\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}})^T \begin{pmatrix} \hat{p}_{1,1}-\mu_{\hat{p}_{1,1}} \\ \vdots \\ \hat{p}_{N,n_N}-\mu_{\hat{p}_{N,n_N}} \end{pmatrix} \\
&+\frac{1}{2}\begin{pmatrix} \hat{p}_{1,1}-\mu_{\hat{p}_{1,1}} \\ \vdots \\ \hat{p}_{N,n_N}-\mu_{\hat{p}_{N,n_N}} \end{pmatrix}^T \nabla^2\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}}) \begin{pmatrix} \hat{p}_{1,1}-\mu_{\hat{p}_{1,1}} \\ \vdots \\ \hat{p}_{N,n_N}-\mu_{\hat{p}_{N,n_N}} \end{pmatrix}
\end{aligned} \tag{5.29}
$$

Before applying the Delta method, a reduction of terms in Eq. (5.29) can be performed. Since we consider systems composed of indepent items with mutually exclusive states, the second derivative of $\hat{r}_k$ with respect to the same item probabilities $\hat{p}_{j,i}$, $i \in \{1,\cdots,n_j\}$, is always zero in our model (due to there is not logically repeated items). Namely, the function $\hat{r}_k$ is a sum of linear products of the item probabilities $\hat{p}_{j,k_m}$, $j \in \{1,\cdots,N\}$, at a unique state k_m for every item j.

$$\frac{\partial^2}{\partial \hat{p}_{j,i}\hat{p}_{j,l}}\hat{r}_k = \frac{\partial^2}{\partial \hat{p}_{j,i}\hat{p}_{j,w}}\left[\sum_{m=1}^{M_k}\prod_{j=1}^{N} p_{j,k_m}\right] = 0 \qquad \forall i,l \in \{1,\cdots,n_j\} \tag{5.30}$$

Now, Eq. (5.29) can be only expressed in terms of the first derivatives and the crossed derivatives of $\hat{r}_k$:

$$\hat{r}_k(\hat{p}_{1,1},\cdots,\hat{p}_{N,n_N}) \approx \hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}}) + \sum_{j=1}^{N}\sum_{i=1}^{n_j}\frac{\partial}{\partial \hat{p}_{j,i}}\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}})\left[\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}}\right]$$
$$+\frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} 2\frac{\partial^2}{\partial \hat{p}_{j,i}\hat{p}_{h,l}}\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}})\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})\right] \tag{5.31}$$

We calculate the variance of the second order Taylor series expansion of $\hat{r}_k$ at $(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{1,n_1}},\cdots,\mu_{\hat{p}_{N,n_N}})$ to approximate the variance of the $\hat{r}_k$ function.

$$Var(\hat{r}_k(\hat{p}_{1,1},\cdots,\hat{p}_{N,n_N})) \;=\; E\left[\{\hat{r}_k(\hat{p}_{1,1},\cdots,\hat{p}_{N,n_N}) - E[\hat{r}_k(\hat{p}_{1,1},\cdots,\hat{p}_{N,n_N})]\}^2\right] \tag{5.32}$$

Appendix B describes the calculation of the $Var(\hat{r}_k)$ to obtain the following expression:

$$Var(\hat{r}_k(\hat{p}_{1,1},\cdots,\hat{p}_{N,n_N})) \approx \sum_{j=1}^{N}\sum_{i=1}^{n_j}\left[\frac{\partial}{\partial \hat{p}_{j,i}}\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}})\right]^2\sigma^2_{\hat{p}_{j,i}}$$
$$+2\sum_{j=1}^{N}\sum_{i=1}^{n_j-1}\sum_{w=i+1}^{n_j} Cov(\hat{p}_{j,i},\hat{p}_{j,w})\left[\frac{\partial}{\partial \hat{p}_{j,i}}\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}})\right]\left[\frac{\partial}{\partial \hat{p}_{j,w}}\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}})\right]$$
$$+\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h}\left[\frac{\partial^2}{\partial \hat{p}_{j,i}\hat{p}_{h,l}}\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}})\right]^2\sigma^2_{\hat{p}_{j,i}}\sigma^2_{\hat{p}_{h,l}}$$
$$+2\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=1,h\neq j}^{N}\sum_{l=1}^{n_h-1}\sum_{v=l+1}^{n_h}\sigma^2_{\hat{p}_{j,i}}Cov(\hat{p}_{h,l},\hat{p}_{h,v})$$
$$*\left[\frac{\partial^2}{\partial \hat{p}_{j,i}\hat{p}_{h,l}}\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}})\right]\left[\frac{\partial^2}{\partial \hat{p}_{j,i}\hat{p}_{h,v}}\hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}})\right]$$

$$+2\sum_{j=1}^{N}\sum_{i=1}^{n_j-1}\sum_{w=i+1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h-1}\sum_{v=l+1}^{n_h} Cov(\hat{p}_{j,i},\hat{p}_{j,w})Cov(\hat{p}_{h,l},\hat{p}_{h,v})$$
$$* \left(\left[\frac{\partial^2}{\partial \hat{p}_{j,i}\hat{p}_{h,l}} \hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}}) \right] \left[\frac{\partial^2}{\partial \hat{p}_{j,w}\hat{p}_{h,v}} \hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}}) \right] \right.$$
$$\left. + \left[\frac{\partial^2}{\partial \hat{p}_{j,i}\hat{p}_{h,v}} \hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}}) \right] \left[\frac{\partial^2}{\partial \hat{p}_{j,w}\hat{p}_{h,l}} \hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}}) \right] \right) \quad (5.33)$$

Equation (5.33) is an improved model of the method presented in Eq. (5.17). The Delta method, applying second order Taylor series expansion, approximates the MSS reliability variance better than the classical Delta method (first order Taylor series) as it is shown below.

For a system composed of binary items, the variance of each reliability system state can be approximated by the following expression:

$$Var(\hat{r}_k(\hat{p}_{1,1},\cdots,\hat{p}_{N,n_N})) \approx \sum_{j=1}^{N}\sum_{i=1}^{n_j} \left[\frac{\partial}{\partial \hat{p}_{j,i}} \hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}}) \right]^2 \sigma^2_{\hat{p}_{j,i}}$$
$$+\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} \left[\frac{\partial^2}{\partial \hat{p}_{j,i}\hat{p}_{h,l}} \hat{r}_k(\mu_{\hat{p}_{1,1}},\cdots,\mu_{\hat{p}_{N,n_N}}) \right]^2 \sigma^2_{\hat{p}_{j,i}}\sigma^2_{\hat{p}_{h,l}} \quad (5.34)$$

Now, the variance of the parallel MSS, shown in Figure 5.2, can be approximated applying the Delta method with second order Taylor series expansion (Eq. (5.34)) in the MSS reliabilities presented in Eq. (5.24).

$$\begin{aligned}
Var(\hat{r}_1) &\approx [-(1-\hat{p}_{2,1})]^2\sigma^2_{\hat{p}_{1,1}} + [(1-\hat{p}_{1,1})]^2\sigma^2_{\hat{p}_{2,1}} + [1]^2\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{2,1}} \\
Var(\hat{r}_2) &\approx [-\hat{p}_{2,1}+(1-\hat{p}_{2,1})]^2\sigma^2_{\hat{p}_{1,1}} + [(1-\hat{p}_{1,1})-\hat{p}_{1,1}]^2\sigma^2_{\hat{p}_{2,1}} + [-2]^2\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{2,1}} \\
Var(\hat{r}_3) &\approx [\hat{p}_{2,1}]^2\sigma^2_{\hat{p}_{1,1}} + [\hat{p}_{1,1}]^2\sigma^2_{\hat{p}_{2,1}} + [1]^2\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{2,1}} \quad (5.35)
\end{aligned}$$

The MSS reliability variance is obtained replacing the correspondent probabilities and variances of every item in Eq. (5.35).

$$\begin{aligned}
Var(\hat{r}_1) &\approx 1.4661E-02 \\
Var(\hat{r}_2) &\approx 1.1844E-02
\end{aligned}$$

$$Var(\hat{r}_3) \approx 2.6100E-04$$

The approximation of the MSS reliability variance obtained by the Delta Method agrees to the simulation results presented in Table 5.2. Besides, the accuracy in the approximation is improved in all the states of the parallel system. This improvement is also attained in other system structures.

5.5 Delta-UGF Method: Computational Assessment

There are systems that possess complex structure functions, which are not easy to calculate their derivatives. Although, there are numerical methods to calculate the numerical derivatives of a function $f(X)$, these methods possess an approximation error. Besides, the application of a numerical method implies an additional computational effort in the MSS reliability variance assessment.

Fortunately, an algorithm can be developed to apply the Delta method in a simple and easy way. The variances of the MSS reliabilities can be calculated obtaining the first derivatives and the crossed second order derivatives of the $\hat{r}_k$. This can be attained applying an iterative assessment process on the u-function $U(z)$.

The derivative of a function

$$f(X_0, X_1, \cdots, X_N) = \sum_{i=0}^{N} a_i X_i, \qquad X_0 = 1 - X_1 - X_2 - \cdots - X_N \tag{5.36}$$

with respect to a variable X_i, $i \in \{1, \cdots, N\}$, can be obtained setting the derivative variable $X_i = 1$ and the dependent variable $X_0 = -1$. The remaining values X_s, $s \neq \{i, 0\}$ are set to zero.

$$\frac{\partial}{\partial X_i} f(X_0, X_1, \cdots, X_N) = a_i - a_0 \tag{5.37}$$

Similarly, the process can be extended to second order derivatives setting both derivative variables, X_i and X_j $(i \neq j)$, equal to 1. Thereby, the MSS reliability variances can be calculated applying this property in each MSS reliability $\hat{r}_k = U(z)_{g_k}$ as follows:

$$
\begin{aligned}
Var(\hat{r}_k) \approx & \sum_{j=1}^{N}\sum_{i=1}^{n_j}\left[U(z,\hat{p}_{j,i}=1,\hat{p}_{j,0}=-1,\hat{p}_{j,s\neq(i,0)}=0)_{g_k}\right]^2\sigma^2_{\hat{p}_{j,i}} \\
+ & \sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h}\left[U(z,\hat{p}_{j,i}=1,\hat{p}_{j,0}=-1,\hat{p}_{j,s\neq(i,0)}=0,\hat{p}_{h,l}=1,\right. \\
& \left.\hat{p}_{h,0}=-1,\hat{p}_{h,s\neq(l,0)}=0)_{g_k}\right]^2\sigma^2_{\hat{p}_{j,i}}\sigma^2_{\hat{p}_{h,l}} \qquad (5.38)
\end{aligned}
$$

For illustrating the UGF-Delta algorithm, the next u-function $U(z)$ is used

$$
U(z) = \hat{p}_{1,0}\hat{p}_{2,0}z^{1.0} + \hat{p}_{1,0}\hat{p}_{2,1}z^{1.5} + \left[\hat{p}_{1,0}\hat{p}_{2,2} + \hat{p}_{1,1}(\hat{p}_{2,0}+\hat{p}_{2,1}+\hat{p}_{2,2})\right]z^{\infty} \qquad (5.39)
$$

This u-function corresponds to the series MSS shown in Figure 5.1. The u-function $U(z)$ can be expressed as function of the estimated probabilities $\hat{p}_{1,1}$, $\hat{p}_{2,1}$ and $\hat{p}_{2,2}$. The variances of the MSS reliabilities can be obtained as it is shown in Eq. (5.38).

First derivatives of $U(z)$:

$$
\begin{aligned}
\frac{\partial}{\partial\hat{p}_{1,1}}U(z) = & \; U(z,\hat{p}_{1,1}=1,\hat{p}_{1,0}=-1) = \underbrace{(-1)\hat{p}_{2,0}}_{\frac{\partial\hat{r}_1}{\partial\hat{p}_{1,1}}}z^{1.0} + \underbrace{(-1)\hat{p}_{2,1}}_{\frac{\partial\hat{r}_2}{\partial\hat{p}_{1,1}}}z^{1.5} \\
+ & \underbrace{\left[(-1)\hat{p}_{2,2}+(1)(\hat{p}_{2,0}+\hat{p}_{2,1}+\hat{p}_{2,2})\right]}_{\frac{\partial\hat{r}_3}{\partial\hat{p}_{1,1}}}z^{\infty} \qquad (5.40) \\
\frac{\partial}{\partial\hat{p}_{2,1}}U(z) = & \; U(z,\hat{p}_{2,1}=1,\hat{p}_{2,0}=-1,\hat{p}_{2,2}=0) = \underbrace{\hat{p}_{1,0}(-1)}_{\frac{\partial\hat{r}_1}{\partial\hat{p}_{2,1}}}z^{1.0} + \underbrace{\hat{p}_{1,0}(1)}_{\frac{\partial\hat{r}_2}{\partial\hat{p}_{2,1}}}z^{1.5} \\
+ & \underbrace{\left[\hat{p}_{1,0}(0)+\hat{p}_{1,1}(-1+1+0)\right]}_{\frac{\partial\hat{r}_3}{\partial\hat{p}_{2,1}}}z^{\infty} \qquad (5.41) \\
\frac{\partial}{\partial\hat{p}_{2,2}}U(z) = & \; U(z,\hat{p}_{2,2}=1,\hat{p}_{2,0}=-1,\hat{p}_{2,1}=0) = \underbrace{\hat{p}_{1,0}(-1)}_{\frac{\partial\hat{r}_1}{\partial\hat{p}_{2,2}}}z^{1.0} + \underbrace{\hat{p}_{1,0}(0)}_{\frac{\partial\hat{r}_2}{\partial\hat{p}_{2,2}}}z^{1.5} \\
+ & \underbrace{\left[\hat{p}_{1,0}(1)+\hat{p}_{1,1}(-1+0+1)\right]}_{\frac{\partial\hat{r}_3}{\partial\hat{p}_{2,2}}}z^{\infty} \qquad (5.42)
\end{aligned}
$$

Second derivatives of $U(z)$:

$$\begin{aligned}
\frac{\partial^2}{\partial \hat{p}_{1,1}\hat{p}_{2,1}}U(z) &= U(z, \hat{p}_{1,1}=1, \hat{p}_{1,0}=-1, \hat{p}_{2,1}=1, \hat{p}_{2,0}=-1, \hat{p}_{2,2}=0) = \underbrace{(-1)(-1)}_{\frac{\partial}{\partial \hat{p}_{2,1}}\left[\frac{\partial \hat{r}_1}{\partial \hat{p}_{1,1}}\right]} z^{1.0} \\
&+ \underbrace{(-1)(1)}_{\frac{\partial}{\partial \hat{p}_{2,1}}\left[\frac{\partial \hat{r}_2}{\partial \hat{p}_{1,1}}\right]} z^{1.5} + \underbrace{[(-1)(0)+(1)(-1+1+0)]}_{\frac{\partial}{\partial \hat{p}_{2,1}}\left[\frac{\partial \hat{r}_3}{\partial \hat{p}_{1,1}}\right]} z^{\infty} \qquad (5.43) \\
\frac{\partial^2}{\partial \hat{p}_{1,1}\hat{p}_{2,2}}U(z) &= U(z, \hat{p}_{1,1}=1, \hat{p}_{1,0}=-1, \hat{p}_{2,2}=1, \hat{p}_{2,0}=-1, \hat{p}_{2,1}=0) = \underbrace{(-1)(-1)}_{\frac{\partial}{\partial \hat{p}_{2,2}}\left[\frac{\partial \hat{r}_1}{\partial \hat{p}_{1,1}}\right]} z^{1.0} \\
&+ \underbrace{(-1)(0)}_{\frac{\partial}{\partial \hat{p}_{2,2}}\left[\frac{\partial \hat{r}_2}{\partial \hat{p}_{1,1}}\right]} z^{1.5} + \underbrace{[(-1)(1)+(1)(-1+0+1)]}_{\frac{\partial}{\partial \hat{p}_{2,2}}\left[\frac{\partial \hat{r}_3}{\partial \hat{p}_{1,1}}\right]} z^{\infty} \qquad (5.44)
\end{aligned}$$

The first and second derivatives of $U(z)$ (Eq. 5.42 and 5.44) provide the partial derivatives of the MSS reliabilities $\hat{r}_k$ required for applying the Delta method to approximate the MSS reliability variances.

$$\begin{aligned}
Var(\hat{r}_1) &\approx [-\hat{p}_{2,0}]^2\sigma^2_{\hat{p}_{1,1}} + [-\hat{p}_{1,0}]^2\sigma^2_{\hat{p}_{2,1}} + [-\hat{p}_{1,0}]^2\sigma^2_{\hat{p}_{2,2}} + [1]^2\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{2,1}} + [1]^2\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{2,2}} \\
&+ 2[-\hat{p}_{1,0}][-\hat{p}_{1,0}]Cov(\hat{p}_{2,1}, \hat{p}_{2,2}) + 2[1][1]\sigma^2_{\hat{p}_{1,1}}Cov(\hat{p}_{2,1}, \hat{p}_{2,2}) \\
Var(\hat{r}_2) &\approx [-\hat{p}_{2,1}]^2\sigma^2_{\hat{p}_{1,1}} + [\hat{p}_{1,0}]^2\sigma^2_{\hat{p}_{2,1}} + [0]^2\sigma^2_{\hat{p}_{2,2}} + [-1]^2\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{2,1}} + [0]^2\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{2,2}} \\
&+ 2[\hat{p}_{1,0}][0]Cov(\hat{p}_{2,1}, \hat{p}_{2,2}) + 2[-1][0]\sigma^2_{\hat{p}_{1,1}}Cov(\hat{p}_{2,1}, \hat{p}_{2,2}) \\
Var(\hat{r}_3) &\approx [-\hat{p}_{2,2}+1]^2\sigma^2_{\hat{p}_{1,1}} + [0]^2\sigma^2_{\hat{p}_{2,1}} + [\hat{p}_{1,0}]^2\sigma^2_{\hat{p}_{2,2}} + [0]^2\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{2,1}} + [-1]^2\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{2,2}} \\
&+ 2[0][\hat{p}_{1,0}]Cov(\hat{p}_{2,1}, \hat{p}_{2,2}) + 2[0][-1]\sigma^2_{\hat{p}_{1,1}}Cov(\hat{p}_{2,1}, \hat{p}_{2,2}) \qquad (5.45)
\end{aligned}$$

Substituting the assumed reliability and variance values used in the series MSS shown in Figure 5.1, we obtain more accurate MSS reliability variances than the presented in Eq. (5.23) corresponding to the Delta method with a first order Taylor series.

$$\begin{aligned}
Var(\hat{r}_1) &\approx 4.4967E-03 \\
Var(\hat{r}_2) &\approx 4.2241E-04
\end{aligned}$$

$$Var(\hat{r}_3) \approx 7.3255E-03$$

For showing the peformance of the UGF-Delta method, the reliability variance of a series-parallel MSS is approximated applying the explained iterative process.

5.6 Reliability Variance of a Series-Parallel MSS

Figure 5.3 shows the production line presented in Chapter 3. The MSS has a series-parallel structure with transmission flow behavior.

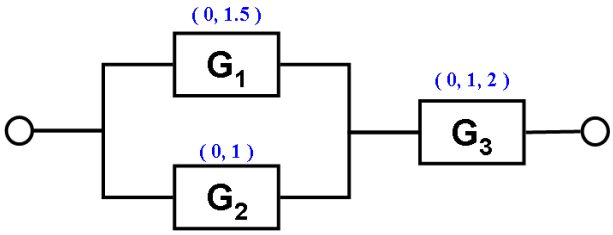


Figure 5.3: Production line

The u-function for every item was presented in Section 3.4.1. Besdies, Section 3.4.1 details how to obtain the u-function $U(z)$ of the entire system.

$$\begin{aligned} U(z) &= \underbrace{[(\hat{p}_{1,1}\hat{p}_{2,1}+\hat{p}_{1,1}\hat{p}_{2,0}+\hat{p}_{1,0}\hat{p}_{2,1}+\hat{p}_{1,0}\hat{p}_{2,0})\hat{p}_{3,2}+\hat{p}_{1,1}\hat{p}_{2,1}\hat{p}_{3,1}+\hat{p}_{1,1}\hat{p}_{2,1}\hat{p}_{3,0}]}_{\hat{r}_1} z^{0.0} \\ &+ \underbrace{[(\hat{p}_{1,1}\hat{p}_{2,0}+\hat{p}_{1,0}\hat{p}_{2,1}+\hat{p}_{1,0}\hat{p}_{2,0})\hat{p}_{3,1}+\hat{p}_{1,1}\hat{p}_{2,0}\hat{p}_{3,0}]}_{\hat{r}_2} z^{1.0} \\ &+ \underbrace{[\hat{p}_{1,0}\hat{p}_{2,1}\hat{p}_{3,0}]}_{\hat{r}_3} z^{1.5} + \underbrace{[\hat{p}_{1,0}\hat{p}_{2,0}\hat{p}_{3,0}]}_{\hat{r}_4} z^{2.0} \end{aligned} \tag{5.46}$$

The reliability variances $Var(\hat{r}_k)$of the series-parallel MSS can be approximated applying the UGF-Delta method with second order Taylor series expansion, as it is illustrated in Eq. (5.38).

First, the state $i = 0$ of each item j is expressed as function of the remaining n_j states.

$$\hat{p}_{1,0} = 1-\hat{p}_{1,1} \qquad \hat{p}_{2,0} = 1-\hat{p}_{2,1} \qquad \hat{p}_{3,0} = 1-\hat{p}_{3,1}-\hat{p}_{3,2} \tag{5.47}$$

Thereby, the derivative variables are $\hat{p}_{1,1}$, $\hat{p}_{2,1}$, $\hat{p}_{3,1}$ and $\hat{p}_{3,2}$. Applying the iterative process explained in Section 5.5 in Eq. (5.46), we obtain the partial derivatives of the MSS reliabilities $\hat{r}_k$, $k \in \{1,2,3,4\}$.

First partial derivatives:

$$
\begin{aligned}
\frac{\partial}{\partial \hat{p}_{1,1}} U(z) &= \underbrace{[(\hat{p}_{2,1} + \hat{p}_{2,0} - \hat{p}_{2,1} - \hat{p}_{2,0})\hat{p}_{3,2} + \hat{p}_{2,1}\hat{p}_{3,1} + \hat{p}_{2,1}\hat{p}_{3,0}]}_{\frac{\partial \hat{r}_1}{\partial \hat{p}_{1,1}}} z^{0.0} \\
&+ \underbrace{[(\hat{p}_{2,0} - \hat{p}_{2,1} - \hat{p}_{2,0})\hat{p}_{3,1} + \hat{p}_{2,0}\hat{p}_{3,0}]}_{\frac{\partial \hat{r}_2}{\partial \hat{p}_{1,1}}} z^{1.0} + \underbrace{[-\hat{p}_{2,1}\hat{p}_{3,0}]}_{\frac{\partial \hat{r}_3}{\partial \hat{p}_{1,1}}} z^{1.5} + \underbrace{[-\hat{p}_{2,0}\hat{p}_{3,0}]}_{\frac{\partial \hat{r}_4}{\partial \hat{p}_{1,1}}} z^{2.0} \\
\frac{\partial}{\partial \hat{p}_{2,1}} U(z) &= \underbrace{[(\hat{p}_{1,1} - \hat{p}_{1,1} + \hat{p}_{1,0} - \hat{p}_{1,0})\hat{p}_{3,2} + \hat{p}_{1,1}\hat{p}_{3,1} + \hat{p}_{1,1}\hat{p}_{3,0}]}_{\frac{\partial \hat{r}_1}{\partial \hat{p}_{2,1}}} z^{0.0} \\
&+ \underbrace{[(-\hat{p}_{1,1} + \hat{p}_{1,0} - \hat{p}_{1,0})\hat{p}_{3,1} - \hat{p}_{1,1}\hat{p}_{3,0}]}_{\frac{\partial \hat{r}_2}{\partial \hat{p}_{2,1}}} z^{1.0} + \underbrace{[\hat{p}_{1,0}\hat{p}_{3,0}]}_{\frac{\partial \hat{r}_3}{\partial \hat{p}_{2,1}}} z^{1.5} + \underbrace{[-\hat{p}_{1,0}\hat{p}_{3,0}]}_{\frac{\partial \hat{r}_4}{\partial \hat{p}_{2,1}}} z^{2.0} \\
\frac{\partial}{\partial \hat{p}_{3,1}} U(z) &= \underbrace{[0 + \hat{p}_{1,1}\hat{p}_{2,1} - \hat{p}_{1,1}\hat{p}_{2,1}]}_{\frac{\partial \hat{r}_1}{\partial \hat{p}_{3,1}}} z^{0.0} \\
&+ \underbrace{[(\hat{p}_{1,1}\hat{p}_{2,0} + \hat{p}_{1,0}\hat{p}_{2,1} + \hat{p}_{1,0}\hat{p}_{2,0}) - \hat{p}_{1,1}\hat{p}_{2,0}]}_{\frac{\partial \hat{r}_2}{\partial \hat{p}_{3,1}}} z^{1.0} + \underbrace{[-\hat{p}_{1,0}\hat{p}_{2,1}]}_{\frac{\partial \hat{r}_3}{\partial \hat{p}_{3,1}}} z^{1.5} + \underbrace{[-\hat{p}_{1,0}\hat{p}_{2,0}]}_{\frac{\partial \hat{r}_4}{\partial \hat{p}_{3,1}}} z^{2.0} \\
\frac{\partial}{\partial \hat{p}_{3,2}} U(z) &= \underbrace{[(\hat{p}_{1,1}\hat{p}_{2,1} + \hat{p}_{1,1}\hat{p}_{2,0} + \hat{p}_{1,0}\hat{p}_{2,1} + \hat{p}_{1,0}\hat{p}_{2,0}) + 0 - \hat{p}_{1,1}\hat{p}_{2,1}]}_{\frac{\partial \hat{r}_1}{\partial \hat{p}_{3,2}}} z^{0.0} \\
&+ \underbrace{[0 - \hat{p}_{1,1}\hat{p}_{2,0}]}_{\frac{\partial \hat{r}_2}{\partial \hat{p}_{3,2}}} z^{1.0} + \underbrace{[-\hat{p}_{1,0}\hat{p}_{2,1}]}_{\frac{\partial \hat{r}_3}{\partial \hat{p}_{3,2}}} z^{1.5} + \underbrace{[-\hat{p}_{1,0}\hat{p}_{2,0}]}_{\frac{\partial \hat{r}_4}{\partial \hat{p}_{3,2}}} z^{2.0} \qquad (5.48)
\end{aligned}
$$

Second partial derivatives:

$$
\begin{aligned}
\frac{\partial^2}{\partial \hat{p}_{1,1}\hat{p}_{2,1}} U(z) &= \underbrace{[(1 - 1 - 1 + 1)\hat{p}_{3,2} + \hat{p}_{3,1} + \hat{p}_{3,0}]}_{\frac{\partial}{\partial \hat{p}_{2,1}}\left[\frac{\partial \hat{r}_1}{\partial \hat{p}_{1,1}}\right]} z^{0.0} \\
&+ \underbrace{[(-1 - 1 + 1)\hat{p}_{3,1} - \hat{p}_{3,0}]}_{\frac{\partial}{\partial \hat{p}_{2,1}}\left[\frac{\partial \hat{r}_2}{\partial \hat{p}_{1,1}}\right]} z^{1.0} + \underbrace{[-\hat{p}_{3,0}]}_{\frac{\partial}{\partial \hat{p}_{2,1}}\left[\frac{\partial \hat{r}_3}{\partial \hat{p}_{1,1}}\right]} z^{1.5} + \underbrace{[\hat{p}_{3,0}]}_{\frac{\partial}{\partial \hat{p}_{2,1}}\left[\frac{\partial \hat{r}_4}{\partial \hat{p}_{1,1}}\right]} z^{2.0}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2}{\partial \hat{p}_{1,1}\hat{p}_{3,1}}U(z) &= \underbrace{[0+\hat{p}_{2,1}-\hat{p}_{2,1}]}_{\frac{\partial}{\partial \hat{p}_{3,1}}\left[\frac{\partial \hat{r}_1}{\partial \hat{p}_{1,1}}\right]} z^{0.0} \\
&+ \underbrace{[(\hat{p}_{2,0}-\hat{p}_{2,1}-\hat{p}_{2,0})-\hat{p}_{2,0}]}_{\frac{\partial}{\partial \hat{p}_{3,1}}\left[\frac{\partial \hat{r}_2}{\partial \hat{p}_{1,1}}\right]} z^{1.0} + \underbrace{[\hat{p}_{2,1}]}_{\frac{\partial}{\partial \hat{p}_{3,1}}\left[\frac{\partial \hat{r}_3}{\partial \hat{p}_{1,1}}\right]} z^{1.5} + \underbrace{[\hat{p}_{2,0}]}_{\frac{\partial}{\partial \hat{p}_{3,1}}\left[\frac{\partial \hat{r}_4}{\partial \hat{p}_{1,1}}\right]} z^{2.0} \\
\frac{\partial^2}{\partial \hat{p}_{1,1}\hat{p}_{3,2}}U(z) &= \underbrace{[(\hat{p}_{2,1}+\hat{p}_{2,0}-\hat{p}_{2,1}-\hat{p}_{2,0})+0-\hat{p}_{2,1}]}_{\frac{\partial}{\partial \hat{p}_{3,2}}\left[\frac{\partial \hat{r}_1}{\partial \hat{p}_{1,1}}\right]} z^{0.0} \\
&+ \underbrace{[0-\hat{p}_{2,0}]}_{\frac{\partial}{\partial \hat{p}_{3,2}}\left[\frac{\partial \hat{r}_2}{\partial \hat{p}_{1,1}}\right]} z^{1.0} + \underbrace{[\hat{p}_{2,1}]}_{\frac{\partial}{\partial \hat{p}_{3,2}}\left[\frac{\partial \hat{r}_3}{\partial \hat{p}_{1,1}}\right]} z^{1.5} + \underbrace{[\hat{p}_{2,0}]}_{\frac{\partial}{\partial \hat{p}_{3,2}}\left[\frac{\partial \hat{r}_4}{\partial \hat{p}_{1,1}}\right]} z^{2.0} \\
\frac{\partial^2}{\partial \hat{p}_{2,1}\hat{p}_{3,1}}U(z) &= \underbrace{[0+\hat{p}_{1,1}-\hat{p}_{1,1}]}_{\frac{\partial}{\partial \hat{p}_{3,1}}\left[\frac{\partial \hat{r}_1}{\partial \hat{p}_{2,1}}\right]} z^{0.0} \\
&+ \underbrace{[(-\hat{p}_{1,1}+\hat{p}_{1,0}-\hat{p}_{1,0})+\hat{p}_{1,1}]}_{\frac{\partial}{\partial \hat{p}_{3,1}}\left[\frac{\partial \hat{r}_2}{\partial \hat{p}_{2,1}}\right]} z^{1.0} + \underbrace{[-\hat{p}_{1,0}]}_{\frac{\partial}{\partial \hat{p}_{3,1}}\left[\frac{\partial \hat{r}_3}{\partial \hat{p}_{2,1}}\right]} z^{1.5} + \underbrace{[\hat{p}_{1,0}]}_{\frac{\partial}{\partial \hat{p}_{3,1}}\left[\frac{\partial \hat{r}_4}{\partial \hat{p}_{2,1}}\right]} z^{2.0} \\
\frac{\partial^2}{\partial \hat{p}_{2,1}\hat{p}_{3,2}}U(z) &= \underbrace{[(\hat{p}_{1,1}-\hat{p}_{1,1}+\hat{p}_{1,0}-\hat{p}_{1,0})+0-\hat{p}_{1,1}]}_{\frac{\partial}{\partial \hat{p}_{3,2}}\left[\frac{\partial \hat{r}_1}{\partial \hat{p}_{2,1}}\right]} z^{0.0} \\
&+ \underbrace{[0+\hat{p}_{1,1}]}_{\frac{\partial}{\partial \hat{p}_{3,2}}\left[\frac{\partial \hat{r}_2}{\partial \hat{p}_{2,1}}\right]} z^{1.0} + \underbrace{[-\hat{p}_{1,0}]}_{\frac{\partial}{\partial \hat{p}_{3,2}}\left[\frac{\partial \hat{r}_3}{\partial \hat{p}_{2,1}}\right]} z^{1.5} + \underbrace{[\hat{p}_{1,0}]}_{\frac{\partial}{\partial \hat{p}_{3,2}}\left[\frac{\partial \hat{r}_4}{\partial \hat{p}_{2,1}}\right]} z^{2.0}
\end{aligned} \tag{5.49}
$$

The MSS reliability variances of the entire series-parallel system are:

$$
\begin{aligned}
Var(\hat{r}_1) &\approx (\hat{p}_{2,1}\hat{p}_{3,1}+\hat{p}_{2,1}\hat{p}_{3,0})^2\,\sigma^2_{\hat{p}_{1,1}} + (\hat{p}_{1,1}\hat{p}_{3,1}+\hat{p}_{1,1}\hat{p}_{3,0})^2\,\sigma^2_{\hat{p}_{2,1}} + (1-\hat{p}_{1,1}\hat{p}_{2,1})^2\,\sigma^2_{\hat{p}_{3,2}} \\
&+ (\hat{p}_{3,1}+\hat{p}_{3,0})^2\,\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{2,1}} + (-\hat{p}_{2,1})^2\,\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{3,2}} + (-\hat{p}_{1,1})^2\,\sigma^2_{\hat{p}_{2,1}}\sigma^2_{\hat{p}_{3,2}} \\
Var(\hat{r}_2) &\approx (\hat{p}_{2,0}\hat{p}_{3,0}-\hat{p}_{2,1}\hat{p}_{3,1})^2\,\sigma^2_{\hat{p}_{1,1}} + (-\hat{p}_{1,1}\hat{p}_{3,1}-\hat{p}_{1,1}\hat{p}_{3,0})^2\,\sigma^2_{\hat{p}_{2,1}} \\
&+ (\hat{p}_{1,0}\hat{p}_{2,1}+\hat{p}_{1,0}\hat{p}_{2,0})^2\,\sigma^2_{\hat{p}_{3,1}} + (-\hat{p}_{1,1}\hat{p}_{2,0})^2\,\sigma^2_{\hat{p}_{3,2}} + (-\hat{p}_{3,1}-\hat{p}_{3,0})^2\,\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{2,1}} \\
&+ (-\hat{p}_{2,1}-\hat{p}_{2,0})^2\,\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{3,1}} + (-\hat{p}_{2,0})^2\,\sigma^2_{\hat{p}_{1,1}}\sigma^2_{\hat{p}_{3,2}} + (\hat{p}_{1,1})^2\,\sigma^2_{\hat{p}_{2,1}}\sigma^2_{\hat{p}_{3,2}} \\
&+ 2\,(\hat{p}_{1,0}\hat{p}_{2,1}+\hat{p}_{1,0}\hat{p}_{2,0})\,(-\hat{p}_{1,1}\hat{p}_{2,0})\,Cov(\hat{p}_{3,1},\hat{p}_{3,2}) \\
&+ 2\,(-\hat{p}_{2,1}-\hat{p}_{2,0})\,(-\hat{p}_{2,0})\,\sigma^2_{\hat{p}_{1,1}}Cov(\hat{p}_{3,1},\hat{p}_{3,2}) \\
Var(\hat{r}_3) &\approx (-\hat{p}_{2,1}\hat{p}_{3,0})^2\,\sigma^2_{\hat{p}_{1,1}} + (\hat{p}_{1,0}\hat{p}_{3,0})^2\,\sigma^2_{\hat{p}_{2,1}} + (-\hat{p}_{1,0}\hat{p}_{2,1})^2\,\sigma^2_{\hat{p}_{3,1}} + (-\hat{p}_{1,0}\hat{p}_{2,1})^2\,\sigma^2_{\hat{p}_{3,2}}
\end{aligned}
$$

Table 5.3: Item values for the series-parallel MSS

Item	**State**	**Peformance**	$\hat{p}_{j,i}$
1	0	1.5	0.80
	1	0.0	0.20
2	0	1.0	0.90
	1	0.0	0.10
3	0	2.0	0.70
	1	1.0	0.20
	2	0.0	0.10

$$
\begin{aligned}
&+ \left(-\hat{p}_{3,0}\right)^2 \sigma^2_{\hat{p}_{1,1}} \sigma^2_{\hat{p}_{2,1}} + \left(\hat{p}_{2,1}\right)^2 \sigma^2_{\hat{p}_{1,1}} \sigma^2_{\hat{p}_{3,1}} + \left(\hat{p}_{2,1}\right)^2 \sigma^2_{\hat{p}_{1,1}} \sigma^2_{\hat{p}_{3,2}} + \left(-\hat{p}_{1,0}\right)^2 \sigma^2_{\hat{p}_{2,1}} \sigma^2_{\hat{p}_{3,1}} \\
&+ \left(-\hat{p}_{1,0}\right)^2 \sigma^2_{\hat{p}_{2,1}} \sigma^2_{\hat{p}_{3,2}} + 2\left(-\hat{p}_{1,0}\hat{p}_{2,1}\right)\left(-\hat{p}_{1,0}\hat{p}_{2,1}\right) Cov(\hat{p}_{3,1}, \hat{p}_{3,2}) \\
&+ 2\left(\hat{p}_{2,1}\right)\left(\hat{p}_{2,1}\right) \sigma^2_{\hat{p}_{1,1}} Cov(\hat{p}_{3,1}, \hat{p}_{3,2}) + 2\left(-\hat{p}_{1,0}\right)\left(-\hat{p}_{1,0}\right) \sigma^2_{\hat{p}_{2,1}} Cov(\hat{p}_{3,1}, \hat{p}_{3,2}) \\
Var(\hat{r}_4) \approx\ & \left(-\hat{p}_{2,0}\hat{p}_{3,0}\right)^2 \sigma^2_{\hat{p}_{1,1}} + \left(-\hat{p}_{1,0}\hat{p}_{3,0}\right)^2 \sigma^2_{\hat{p}_{2,1}} + \left(-\hat{p}_{1,0}\hat{p}_{2,0}\right)^2 \sigma^2_{\hat{p}_{3,1}} + \left(-\hat{p}_{1,0}\hat{p}_{2,0}\right)^2 \sigma^2_{\hat{p}_{3,2}} \\
&+ \left(\hat{p}_{3,0}\right)^2 \sigma^2_{\hat{p}_{1,1}} \sigma^2_{\hat{p}_{2,1}} + \left(\hat{p}_{2,0}\right)^2 \sigma^2_{\hat{p}_{1,1}} \sigma^2_{\hat{p}_{3,1}} + \left(\hat{p}_{2,0}\right)^2 \sigma^2_{\hat{p}_{1,1}} \sigma^2_{\hat{p}_{3,2}} + \left(\hat{p}_{1,0}\right)^2 \sigma^2_{\hat{p}_{2,1}} \sigma^2_{\hat{p}_{3,1}} \\
&+ \left(\hat{p}_{1,0}\right)^2 \sigma^2_{\hat{p}_{2,1}} \sigma^2_{\hat{p}_{3,2}} + 2\left(-\hat{p}_{1,0}\hat{p}_{2,0}\right)\left(-\hat{p}_{1,0}\hat{p}_{2,0}\right) Cov(\hat{p}_{3,1}, \hat{p}_{3,2}) \\
&+ 2\left(\hat{p}_{2,0}\right)\left(\hat{p}_{2,0}\right) \sigma^2_{\hat{p}_{1,1}} Cov(\hat{p}_{3,1}, \hat{p}_{3,2}) + 2\left(\hat{p}_{1,0}\right)\left(\hat{p}_{1,0}\right) \sigma^2_{\hat{p}_{2,1}} Cov(\hat{p}_{3,1}, \hat{p}_{3,2}) \qquad (5.50)
\end{aligned}
$$

For demonstrating the accuracy of the approximated $Var(U(z))$, a simulation of the series-parallel MSS was perfomed. Table 5.3 provides the probabilities assumed for every item stated.

The following variance matrix is assumed for every item:

$$
\Sigma_1 = \begin{bmatrix} 4.00E-04 & -4.00E-04 \\ -4.00E-04 & 4.00E-04 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 2.25E-04 & -2.25E-04 \\ -2.25E-04 & 2.25E-04 \end{bmatrix}
$$

$$
\Sigma_3 = \begin{bmatrix} 1.00E-04 & -4.67E-05 & -2.33E-05 \\ -4.67E-05 & 7.62E-05 & -6.67E-06 \\ -2.33E-05 & -6.67E-06 & 4.29E-05 \end{bmatrix} \qquad (5.51)
$$

Table 5.4 presents the results of $1,000,000$ simulations of the entire system. The items were simulated applying the following Dirichlet distributions: Item1 $\sim Dir(319.2, 79.8)$, Item2 $\sim Dir(359.1, 39.9)$, and Item3 $\sim Dir(1469.3, 419.8, 209.9)$.

Table 5.4 provides the reliability variance obtained by the UGF-Delta method. The series-parallel MSS example proves the accuracy of the UGF-Delta method approximation with respect

Table 5.4: Reliability simulations of the parallel MSS with work sharing

			Reliability			Variance	
Entity	**State**	**Peformance**	**Min**	**Max**	**Mean**	**Simulation**	**UGF-Delta**
Item 1	0	1.5	0.699225	0.887445	0.799998	3.9984E-04	4.0000E-04
	1	0.0	0.112555	0.300775	0.200002	3.9984E-04	4.0000E-04
Item 2	0	1.0	0.817947	0.959192	0.900003	2.2514E-04	2.2500E-04
	1	0.0	0.040808	0.182053	0.099997	2.2514E-04	2.2500E-04
Item 3	0	2.0	0.653981	0.744241	0.700014	1.0010E-04	1.0000E-04
	1	1.0	0.161896	0.241623	0.199989	7.6293E-05	7.6200E-05
	2	0.0	0.070443	0.134919	0.099997	4.2890E-05	4.2900E-05
MSS	1	0.0	0.086996	0.156871	0.117997	5.1844E-05	5.1805E-05
	2	1.0	0.256226	0.397308	0.321993	2.0934E-04	2.0833E-04
	3	1.5	0.022678	0.105726	0.055999	7.3232E-05	7.3257E-05
	4	2.0	0.420305	0.582658	0.504011	2.8161E-04	2.8424E-04

to the simulation results.

In this chapter, the Delta method is applied to propagate the variance of the item probabilities. There is no assumed time-to-failure distribution of the items or system. This is very practical when item or system failure time is difficult to record or few failures are available to specify an appropriate distribution. The UGF-Delta method can be applied to the systems presented in Chapter 2 or complex combinations of themselves.

Chapter 6

Particle Evolutionary Swarm Optimization

Reliability engineering involves the optimal design of complex systems for finding the suitable tradeoff between reliability, cost and performance.

The diversity of system structures, resource constraints, and options for reliability improvement has led to the construction and analysis of several optimization models. We are interested in the two general *redundancy allocation* and *reliability allocation* problems. For the first, there are discrete item choices with known characteristics (reliability, cost, weight, etc.). The objective for this combinatorial problem is to decide which items to use and the corresponding redundancy levels. The redundancy allocation problem has been shown to be NP-hard [9]. For the second, item's reliability (or a vector of distribution parameters) is treated as a design variable, and the item's cost is a function of its reliability.

Finding an optimal balance between these two factors is a classical reliability-redundancy allocation problem (RRAP) [77]. Two types of decision variables are of concern in the RRAP: real numbers representing probabilities of the items, and integer numbers representing the number of items in each subsystem [77].

The RRAP deals with the optimization of reliability and cost. The optimization criterion could appear of two different forms:

- Maximization of the system reliability, subject to cost constraints.
- Minimization of the system cost, subject to reliability constraints.

Several optimization approaches have previously been used to solve the reliability allocation problem [61]. Evolutionary algorithms have been widely used to solve the RRAP [12, 20, 77, 124, 125]. Mainly, genetic algorithms (GAs) have been used in many works for solve reliability optimization problems [67, 83, 125]. However, the tests performed in [20] showed that the particle swarm optimization algorithm (PSO) is able to get better results than those obtained by a GA for several reliability optimization problems.

Evolutionary algorithms are unconstrained search techniques which lack an explicit mechanism to deal with constrained search spaces. Therefore, introducing new good optimization approaches can be very helpful in some specific areas and can benefit practitioners by providing more choices [77]. This thesis proposes to apply a modified version of the PSO algorithm to solve design MSS reliability problems.

6.1 Particle Swarm Optimization

Particle swarm optimization (PSO) algorithm is a population-based optimization technique inspired by the motion of a bird flock, or fish schooling. In the PSO model, every particle flies over a real valued n-dimensional space of decision variables X. Each particle keeps track of its position X, velocity V, and remembers the best position ever visited, P_{Best}. The particle with the best P_{Best} value is called the leader, and its position is called global best, G_{Best}. The next particle's position is computed by adding a velocity term to its current position, as follows:

$$X_{t+1} = X_t + V_{t+1} \tag{6.1}$$

The velocity term combines the local information of the particle with global information of the flock, in the following way.

$$V_{t+1} = w * V_t + \phi_1 * (P_{Best} - X_t) + \phi_2 * (G_{Best} - X_t) \tag{6.2}$$

The equation above reflects the socially exchanged information. It summarizes PSO three main features: distributed control, collective behavior, and local interaction with the environment [21, 57].The second term is called the cognitive component, while the last term is called the social component. w is the inertia weight, and ϕ_1 and ϕ_2 are called acceleration coefficients. The inertia weight regulates the mixture of the previous velocity with the current one.

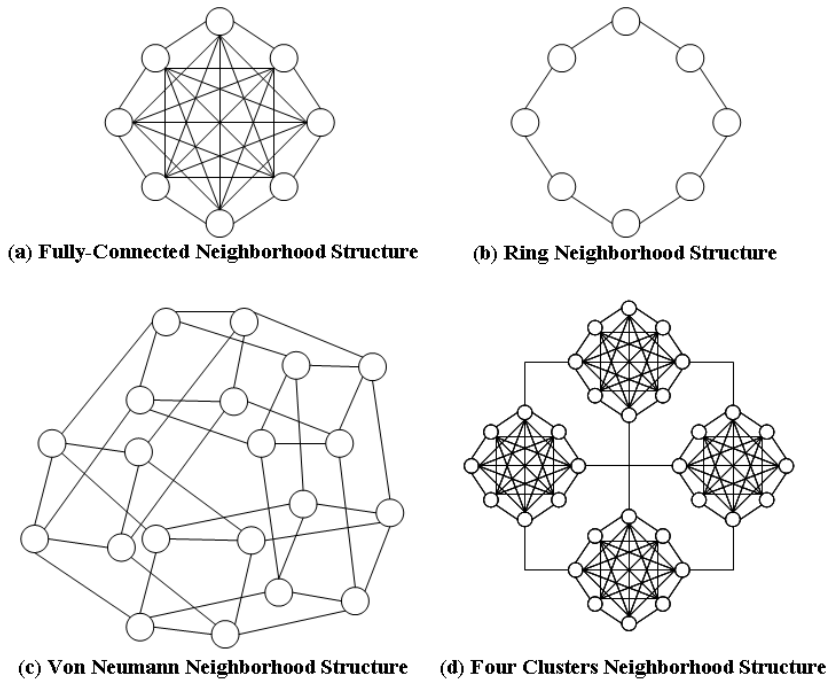


Figure 6.1: Neighborhood structures for PSO

When the flock is split into several neighborhoods the particle's velocity is computed with respect to its neighbors. The best P_{Best} value in the neighborhood is called the local best, L_{Best}.

$$V_{t+1} = w * V_t + \phi_1 * (P_{Best} - X_t) + \phi_2 * (L_{Best} - X_t) \tag{6.3}$$

Neighborhoods can be interconnected in many different ways. Some of the most popular interconnection neighborhoods are shown in Figure 6.1. The star topology is in fact one big neighborhood where every particle is connected to each other, thus enabling the computation of a global best. The ring topology allows neighborhoods and it is, therefore, commonly used by PSO 's local best model.

PSO is a fast algorithm whose natural behavior is to quickly converge to the best explored local optima. However, attainting flock's convergence to the global optimum with high probability implies certain adaptations. The approaches range from modifications to the main PSO equation, to the incorporation of reproduction operators. This chapter introduces the Particle Evolutionary Swarm Optimization algorithm (PESO), a hybrid PSO with the following

new features:

- Two perturbation operators to keep diversity. Although perturbations are not included in the original PSO model, they are quite common nowadays. The formal analysis of van den Bergh shows that the PSO algorithm will only converge to the best position visited, not the global optimum [132]. Therefore, the quest for high diversity is a sound approach to locate the global optimum since more diversity leads to increase the exploratory capabilities of PSO.

- Singly linked ring topology. PESO creates several neighborhoods around a new ring topology that we called the "singly linked ring topology" (SLR). Thus PESO promotes a flock with several local leaders to improve the exploration capacity [21, 57]).

- Constraint handling. PSO lacks an explicit mechanism to bias the search towards the feasible region in constrained search spaces. For selecting a neighborhood leader, PESO picks the winner through a tournament of feasibility and sum of constraint violations ("superiority of feasible solutions" [18]).

6.2 Diversity control for PSO algorithm

A frequent problem in evolutionary computation is premature convergence. This means that the evolutionary algorithm could be trapped in a region containing a local optimum. Premature convergence can be caused by the loss of diversity, which occurs when the population reaches a suboptimal state where the evolutionary algorithm can no longer produce offspring which outperform their parents [27]. A way to keep diversity is by maintaining a proper balance between exploration and exploitation [39].

In the standard PSO, the source of diversity, called *variation*, is measured as the differences between: 1)the particle's position X and the particle's memory P_{Best}, and 2)the particle's position X and the best particle's neighbor L_{Best}. Although variation provides diversity, it can only be maintained for a limited number of generations because convergence of the flock is necessary to refine the solution.

The need to keep diversity is a well-known topic in PSO research. It has been addressed by several authors who have developed many variants, which are summarized (for the purposes of this dissertation) in five categories:

1. **Modified velocity term**: the formulation of the velocity update rule is modified in one or all of its terms (inertia, cognitive and social). García-Villoria and Pastor proposed to introduce diversity into a discrete PSO algorithm by adding a random velocity [28]. This mechanism changes dynamically the degree of the introduced diversity according to the heterogeneity of the population. Jie and Zeng introduced a negative feedback mechanism into PSO [48]. Through the feedback control of the diversity, their approach can manipulate the weight of the cognitive part and the social part to fluctuate with the search state. Das et al. [16] adapted the reproduction operator of differential evolution [122, 123] to the PSO velocity term.

2. **Swarm breeding**: a breeding operator (reproduction, recombination, crossover, etc.) is incorporated into the standard PSO to generate offspring particles benefited from two or more parents. Lovbjerg et al. proposed a standard PSO combined with arithmetic crossover (real valued GA crossover) [85]. Settles and Soule proposed a new crossover operator, called VPAC, which incorporates the PSO velocity vector [120]. The VPAC operator creates two offspring particles whose positions are between the parents positions, but are accelerated away from their current parents' direction.

3. **Evolutionary combinations**: the PSO algorithm is improved with the incorporation of other evolutionary algorithms procedures every k generations. Zhang and Xie proposed to compute the velocity term by taking turns between PSO and differential evolution [143]. At odd generations the individuals are updated by the rules of motion of PSO; at even generations by the differential evolution formalism.

4. **Clusters**: the approaches measure and control the swarm crowding keeping away particles among themselves. For instance, the SEPSO proposed by Krink et al., involves endowing each particle with a radius, then causing particles to bounce off of one another [59]. Blackwell introduces the concept of "charged" particles based on an electrostatic analogy-inter-particle repulsions [4]. A related approach, called ARPSO, measures the global diversity of the swarm, triggering modes of global attraction or repulsion when it crosses predefined thresholds [116].

PESO keeps diversity by applying two perturbation operators on P_{Best}: the C-perturbation which is oriented to sustain global exploration by keeping diversity, and the M-perturbation oriented to the local refinement of solutions. Angeline [2], and also Eberhart and Shi [22], proposed swarm breeding as a good mecanishm to maintain PSO diversity.

Since the L_{Best} is a copy of the P_{Best} memory of the best neighbor particle, it is a good idea to perturb the particle's memory P_{Best} to maintain population diversity; due to the source of the PSO diversity is related with the differences between vectors X and P_{Best}, and between vectors X and L_{Best}.

6.3 Constraint handling techniques for the PSO algorithm

Real optimization problems are subject to a number of equality and inequality constraints, which can be linear or nonlinear. These constraints determine which areas of the search space are feasible and which are infeasible. In addition to these constraints, boundary constraints are usually imposed to the search space (Michalewicz, 1992). Also, there is the possibility that the feasible space is fragmented and separated by infeasible regions, requiring that both the feasible and infeasible regions be searched.

Several authors have noted how the adoption of a constraint handling technique may cause diversity loss due to the additional selection pressure required to bias the population towards the feasible region [34, 36, 89, 37].

PSO is an unconstrained search technique. Thus, adopting a constraint handling technique into the main PSO algorithm is an open research area. There is a considerable amount of research regarding mechanisms that allow the PSO algorithm to deal with equality and inequality constraints. Several constraint-handling techniques have been imported from traditional methods and from the evolutionary literature [11], while others have been developed by exploiting specific features of the PSO algorithm [46].

Different classifications can be found in the literature to group somehow similar techniques, although it is not always straightforward to decide which technique belongs to which group [91, 11, 23, 46]. In general, the literature arranges the constraint-handling methods into five categories:

1. **Preserving only feasible solutions**: This method requires successive random initialization until a feasible solution or set of feasible solutions is generated. Hu and Eberhart proposed a feasibility preservation strategy that determines the best particle [41, 42]. While particles can fly over infeasible regions, they are quickly pulled back to feasible space as infeasible positions are not stored in memory.

2. **Penalty functions methods**: These methods transform the original constrained

problem into an unconstrained one by penalizing the objective function associated with infeasible solutions. The penalty function is very sensitive to the tuning of the penalty coefficients. For instance, Parsopoulos and Vrahatis used a multi-stage assignment penalty function without diversity control [108].

3. **Superiority of feasible solutions**: This method compares two feasible solutions based on the objective function values (fitness). Between a feasible solution and an infeasible solution, a feasible solution is always chosen rather than an infeasible one. When comparing two infeasible solutions, constraint violations information is used as a criterion to choose the best one. For instance Toscano and Coello [128], use a turbulence operator (a kind of mutation) combined with a *superiority of feasible solutions* method [18].

4. **Repair methods**: In these methods, some procedures are applied so that an infeasible candidate solution is repaired by relocating it to a nearby feasible position. For example, He et al. proposed a "fly-back" mechanism that returns an unfeasible particle to its previous feasible position [35].

5. **Multi-objective based methods**: The main idea is to think of constraint violations as additional objectives to be minimized. Thus, constrained single-objective problems can be tackled using multi-objective optimization techniques [90].

Early approaches did not combine a diversity maintenance strategy with the constraint handling technique. For instance, Coath and Halgamuge [10] analyzed PSO algorithms based on penalty and feasibility preservation methods. Their experiments clearly detect the need of some form of diversity control. PESO uses a superiority of feasible solutions method to handle constraints and additionally applies a *dynamic tolerance* technique for handling equality constraints [97].

6.4 The PESO Algorithm

PESO is a local PSO with a `singly-linked` ring neighborhood. PESO handles constraints by adopting a *superiority of feasible solutions* method complemented with a *dynamic tolerance* (for handling equality constraints). The main components of PESO are the *C-Perturbation* and *M-Perturbation* operators; these are applied to the variable P_{Best} of the flock [92].

Throughout the current section, the essential components of PESO are explained: interaction model, neighborhood structure, diversity mechanism and constraint-handling mechanism.

6.4.1 Interaction Model

There are 4 interaction models proposed by Kennedy [56]; these models were defined by omitting components of the velocity formula. The *full* model is composed by the cognition component and the social component. Dropping the social component results in the *cognition-only* model, whereas dropping the cognition component defines the *social-only* model. In a fourth model, *selfless* model, the neighbourhood best is chosen only from the neighbours, without considering the current individual. Experimental results have shown that the social-only model consistently found solutions faster than the full model, but the reliability of the social-only model is lower than the full model.

Parsopoulos and Vrahatis [109] proposed a new scheme that combine the global and the local PSO variants. The approach calculates two velocity directions, one from a global full model and another from a local full model. Both directions, global and local, are linearly combined trough an *influence* parameter. In the same way, Cagnina et al. [7] proposed an extended full model using a combination of both the global and the local PSO variants with a constriction factor k (the k parameter affects the whole velocity equation). In PESO, the velocity equation combines a new global term with the inertia, cognition and social terms.

$$V_{t+1} = w * V_t + \phi_1 * (P_{Best} - X_t) + \phi_2 * (L_{Best} - X_t) + \phi_3 * (G_{Best} - X_t) \quad (6.4)$$

The inertia weight in PESO is linearly decremented from 1.0 to 0.5 according to the number of function evaluations. Thus, at the beginning of the search, the algorithm performs more exploration, and along the process the algorithm is gradually focused on exploitation. The experiments suggest that the best parameters for the model are the following:

$$\phi_1 = 1.0 * U(0, 1.0) = U(0, 1.0)$$
$$\phi_2 = 1.0 * U(0, 1.0) = U(0, 1.0)$$
$$\phi_3 = 0.1 * U(0, 1.0) = U(0, 0.1)$$

where $U(\alpha, \beta)$ is a uniform distribution with random numbers between α and β. Note that the new global term in the velocity equation has a low influence (ϕ_3) but its role is nevertheless important to keep the flock on promising regions. This concept will be understood when we

analyze the technique to handle equality constraints, where the incorporation of the global influence is essential to improve the results of some test functions.

6.4.2 Neighborhood Structure

In the PSO scheme each particle moves following a leader. Particles in the same neighborhood communicate with one another by exchanging information for moving towards a better position. The flow of information through the flock depends on the neighborhood structure. Figure 6.1 presents a few neighborhood structures developed for PSO.

In a highly connected neighborhood structure, the information about the best particle in the swarm (G_{Best}) is quickly transmitted through the whole flock. This means faster convergence, but also implies a higher risk to converge to a local minimum. Also, Kennedy and Mendes empirically show that the *star* neighborhood attains faster convergence than the other topologies, but it meets the optimal fewer times than any other one [58]. They suggest trying the *Von Neumann* neighborhood structure, which performed more consistently in their experiments than the topologies commonly found in current practice. The success of the Von Neumann neighborhood in unconstrained optimization is due to the interaction that each particle has with other particles, an average of 5 neighbors. This promotes the exploitation, but unfortunately fails to provide the exploration required by the constrained optimization problems. It is important to note that the conclusions reached by Kennedy and Mendes are valid for unconstrained optimization problems. PESO works with a new neighborhood structure, the *singly-linked ring*.

The singly-linked ring (SLR) neighborhood rises from discovering that the ring and the Von Neumann neighborhoods are double-linked lists; like it is shown in Figure 6.2. Suppose that every particle is assigned a permanent label which is used to construct the neighborhoods. For each particle k, a neighborhood of size n is composed by the next $n/2$ linked particles, and by $n/2$ previous particles; and the best particle in the neighborhood is the local best of particle k (L_{Best}). For example, in a neighborhood of size $n = 2$, particle k has two neighbors, particles $k-1$ and $k+1$. In turn, particles $k-1$ and $k+1$ have particle k as a neighbor. In this way, there is a mutual attraction between consecutive particles, forming overlapped clusters. Also, the slower convergence of the ring structure has been empirically shown [58].

Now let's, analyze a PSO based on a SLR neighborhood, see Figure 6.3, and assume particle k is again the best of flock. But particle k now has particles $k-2$ and $k+1$ its as neighbors (not $k-1$ and $k+1$ as in the double link). Since particle $k+1$ has particles $k-1$ and $k+2$

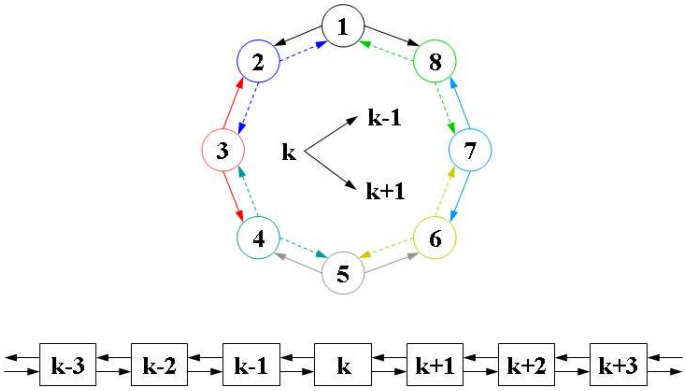


Figure 6.2: Ring structure

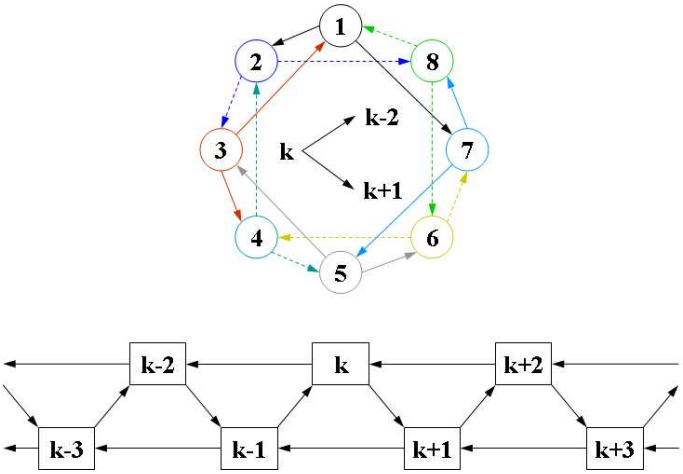


Figure 6.3: Singly-linked ring structure

as its neighbors, and $k-1$ has particles $k-3$ and k as its neighbors. Then, k attracts $k-1$ but $k-1$ only attracts k through particle $k+1$. Therefore, the particle in between cancels the mutual attraction. Besides, the information through the whole swarm is transmitted faster than in the original ring topology. Therefore, the SLR structure keeps the exploration of the search space, and increases the exploitation of the best solutions [38].

Algorithm 1 shows the procedure to find the neighbors for particle k in a SLR structure; where N_k is the neighborhood of particle k, and $P_{(k+m)}$ is the particle located m positions beyond particle k.

The procedure presented in Algorithm 1 allows neighborhoods of size $N = flock/2 - 1$ without mutual attraction. In neighborhoods of size $N = flock/2$ appears a double link between particles k and $k + flock/2$. PESO uses a SLR structure with a neighborhood of size $n = 4$.

Algorithm 1 Singly-linked ring neighboorhood

$N_k = \emptyset$
$Step = 1$
$Switch = 1$
repeat
  $N_k = N_k \cup P_{(k+Switch*Step)}$
  $Step = Step + 1$
  $Switch = -1 * Switch$
until $N_k = N$

For each particle k, its neighbors are the particles: $k+1$, $k-2$, $k+3$ and $k-4$. A detailed explanation of the SLR structure with 4 neighbors is given in Table 6.1.

Table 6.1 shows the neighbors for every particle k and the information flow on the communications channels (neighbors) spawn by the proposed topology.

6.4.3 Diversity Mechanism

In his dissertation, van den Bergh gives a theorem which specifies under which conditions an algorithm can be considered a global optimization method [132]. The theorem implies that a general algorithm, without *a priori* knowledge, must be able to generate an infinite number of samples distributed throughout the whole search space in order to guarantee that it will find the global optimum with asymptotic probability 1.

This can be achieved by periodically adding randomized particles to the swarm. Nevertheless, resetting the position of the particles is not a trivial task; a bad decision affects directly in the exploitation of the best solutions [132]. We propose, based on the observation that the P_{Best} particles drive the swarm, to perturb the P_{Best} population.

Before explaining the perturbation operators, it is necessary to introduce the main algorithm of PESO for better understanding. The complete pseudocode of the PESO algorithm is shown in Figure 2.

Flying the particles is the main task of PSO, see Stage 1. Variables LL and UL are the lower and upper limits of the search space. Function $LocalBest$ returns the best neighbor, applying a SLR structure with 4 neighbors, for each particle. Function $ParticleBest$ updates the P_{Best} population. Function $TheBest$ updates the G_{Best} particle. The function SCV computes the Sum of Constraint Violations, that is, total value of infeasible constraints. The number of particles is n, d is the dimension of the space, and i is the generation index. The perturbations

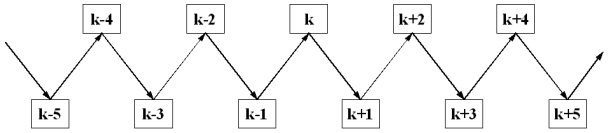
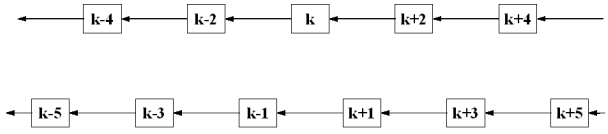
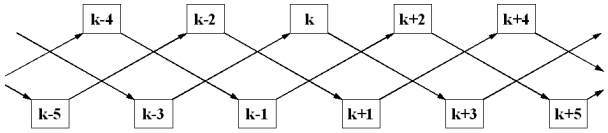
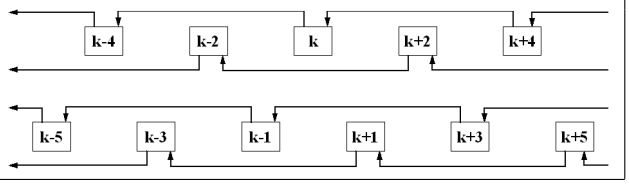
N	Neighbor	Structure
1	$k+1$	
2	$k-2$	
3	$k+3$	
4	$k-4$	

Table 6.1: Singly-Linked Ring Structure with $n = 4$ neighbors

Algorithm 2 Pseudo-code of the $PESO$ algorithm

$X_0 = \text{Rand}(LL, UL)$
$V_0 = 0$
$F_0 = \text{Fitness}\ (\ X_0\ )$
$C_0 = \text{SCV}\ (\ X_0\ )$
$P_{Best} = X_0, \quad F_{Best} = F_0, \quad C_{Best} = C_0$
for $i = 1$ to $maxgenerations$ **do**
  Stage 1
  $L_{Best} =$ **LocalBest** (F_{Best}, C_{Best})
  $G_{Best} =$ **TheBest** (F_{Best}, C_{Best})
  $V =$ **Velocity** (V, X, P_{Best}, L_{Best}, G_{Best})
  $X = X + V$
  $F = \text{Fitness}\ (\ X\ )$
  $C =$ **SCV** (X)
  $[P_{Best}, F_{Best}, C_{Best}] =$ **ParticleBest** (P_{Best}, X, F_{Best}, F, C_{Best}, C)
  Stage 2
  if $U(0,1) < p_C$ **then**
    $P_{Temp} =$ **C-Perturbation** (P_{Best})
    $F_{Temp} = \text{Fitness}\ (\ P_{Temp}\ )$
    $C_{Temp} =$ **SCV** (P_{Temp})
    $[P_{Best}, F_{Best}, C_{Best}] =$ **ParticleBest** (P_{Best}, P_{Temp}, F_{Best}, F_{Temp}, C_{Best}, C_{Temp})
    $T_{Best} =$ **TheBest** (F_{Best}, C_{Best})
    if $T_{Best} \geq G_{Best}$ **then**
      $p_C = p_C$*0.99
    else
      $p_C = \min(1.0, p_C$*1.01)
      $G_{Best} = T_{Best}$
    end if
  end if
  Stage 3
  if $U(0,1) < p_C$ **then**
    $P_{Temp} =$ **M-Perturbation** (P_{Best})
    $F_{Temp} = \text{Fitness}\ (\ P_{Temp}\ )$
    $C_{Temp} =$ **SCV** (P_{Temp})
    $[P_{Best}, F_{Best}, C_{Best}] =$ **ParticleBest** (P_{Best}, P_{Temp}, F_{Best}, F_{Temp}, C_{Best}, C_{Temp})
    $T_{Best} =$ **TheBest** (F_{Best}, C_{Best})
    if $T_{Best} \geq G_{Best}$ **then**
      $p_M = p_M$*0.99
    else
      $p_M = \min(1.0, p_M$*1.01)
      $G_{Best} = T_{Best}$
    end if
  end if
end for

are applied to P_{Best} in the next two stages.

The perturbation of P_{Best} is what makes PESO different. Other approaches perturb the position of the particle and later P_{Best} is updated accordingly. PESO, however, applies the perturbations to the memory P_{Best} and leaves the motion of the particles to PSO.

The goal of the second stage is to add a perturbation generated from the linear combination of three random vectors. This perturbation is preferred over other operators because it preserves the distribution of the population (also used for reproduction by the differential evolution algorithm [111]). In PESO this perturbation is called C-Perturbation. It is applied to the members of PBest to yield a set of temporal particles P_{Temp}. Then each member of P_{Temp} is compared with its corresponding father and P_{Best} is updated with the child if it wins the tournament. Algorithm 3 shows the pseudo-code of the **C-Perturbation** operator.

Algorithm 3 Pseudo-code of **C-Perturbation**

```
for k = 0 to n do
  for j = 0 to d do
    r = U(0, 1)
    p1 = Random(n)
    p2 = Random(n)
    Temp[k, j] = PBest[k, j] + r (PBest[p1, j] - PBest[p2, j])
  end for
end for
```

In the third stage every vector is perturbed again, so that a particle can be deviated from its current direction as responding to external, perhaps more promissory, stimuli. This perturbation is implemented by adding small random numbers (from a uniform distribution) to every design variable. The perturbation, called M-Perturbation, is applied to every member of PBest to yield a set of temporal particles P_{Temp}. Then each member of P_{Temp} is compared with its corresponding father and P_{Best} is updated with the child if it wins the tournament. Algorithm 4 shows the pseudo-code of the **M-Perturbation** operator. The perturbation is added to every dimension of the decision vector with probability $1/d$ (d is the dimension of the decision variable vector).

In Figure 6.4, the position P_{Best} is relocated to a new "best" after the perturbation operations. Notice that this change is made to the particle's best visited location. When PSO takes its turn to perform its computations, it finds everything as left in the previous generation, except that the memory P_{Best} may now store a better position.

The cooperation between PSO and the perturbation operators has been carefully analyzed

Algorithm 4 Pseudo-code of **M-Perturbation**

for $k = 0$ to n **do**
  for $j = 0$ to d **do**
    $r = U(0, 1)$
    if $r \leq 1/d$ **then**
      $Temp[k, j] = \text{Rand}(LL, UL)$
    else
      $Temp[k, j] = P_{Best}[k, j]$
    end if
  end for
end for

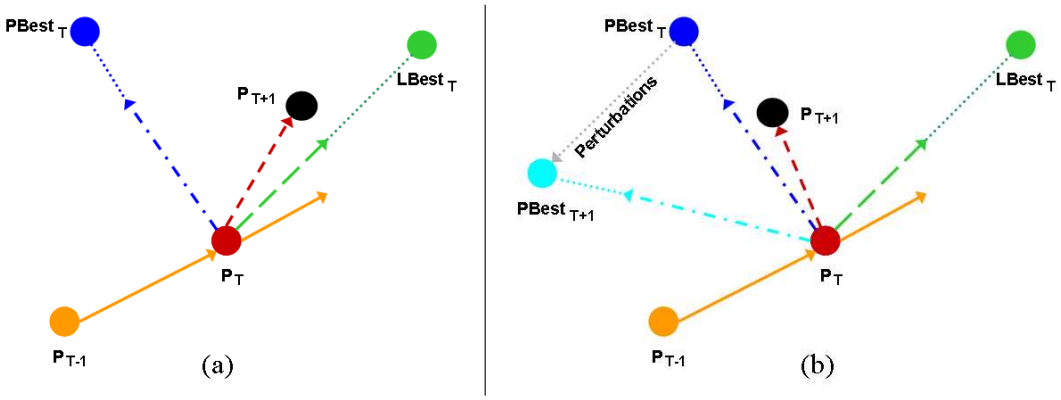


Figure 6.4: PBest updated after the C and M perturbations

by the author throughout the numerous experiments conducted. The PSO stage is very efficient at refining solutions in a local space, but exploration is performed by the perturbation operators. The perturbation operators adapt their activity along generations as follows: the operator is invoked with probability $p_C = p_M = 1$ when the flock is flown for the first time. This probability is decremented by a factor of 1% whenever the perturbation stage fails to improve the G_{Best} value, otherwise, it is incremented by a factor of 1%. Thereby, the incorporation of perturbation to the P_{Best} decreases along the process whenever the swarm converges to the optimal.

In his work, van den Bergh proposed two algorithms which periodically add particles randomly to the flock for accomplishing his theorem that guarantees convergence to the global optimum [132]. Following that line in PESO, a P_{Temp} population is created after applying the perturbation operators, and then used (as if they were random particles) to update the P_{Best} population. The perturbation of the P_{Best} population is performed every generation, subject to the current probabilities p_C and p_M of the perturbation operators. Algorithm 2 shows clearly this feature.

6.4.4 Constraint handling approach

Deb introduced the `superiority of feasible solutions` selection based on the idea that any individual in a constrained search space must first comply with the constraints and then with the function value [18]. PESO adopted such popular tournament selection whose rules are included in the functions $LocalBest$, $ParticleBest$ and $TheBest$:

1. Given two feasible particles, pick the one with better function value;
2. From a pair of feasible and infeasible particles, pick the feasible one;
3. If both particles are infeasible, pick the particle with the lowest sum of constraint violation.

The sum of constraint violations is, of course, the total value by which unsatisfied constraints are violated (computed by function **SCV** in Figure 2). The superiority of feasible solutions method does not require tuning parameters or applying special operators. Just a simple comparison is used to choose the best individual. Our approach applies this method, allowing feasible and infeasible solutions in the swarm. It enriches the information about the search space, especially at its boundaries.

Nevertheless, for handling equality constraints, it is not enough just converting them into inequality constraints of the form $|h_j| \leq \epsilon$, where ϵ is called the tolerance. PESO uses a *dynamic*

tolerance technique for handling equality constraints [97]. An initial tolerance value of $\epsilon = 1.0$ is decremented by 10% to a specified *target value* τ. The tolerance value is decreased whenever a percentage of feasible particles (PFP) is attained. The initial value of PFP is 100%, and it is updated at every generation with the following equation:

$$PFP = \left(1 - \frac{generation}{maxgeneration}\right) \tag{6.5}$$

In addition, when PESO has performed 90% of the function evaluations and the tolerance value has not reached the specified target value ($\epsilon > \tau$), the current tolerance value is replaced with that target value ($\epsilon = \tau$). For the last 10% of the function evaluations, the tolerance value is kept fixed ($\epsilon = \tau$); thus, the particles have additional time to achieve convergence. Summarizing, at the beginning of the process the particles are allowed to be feasible in a large number. Whenever the tolerance is decremented some particles might become infeasible, resulting in the loss of promising regions [97].

6.4.5 Applications of the PESO algorithm

PESO has been applied to solve artificial constrained and unconstrained functions with continuos and discrete variables [93, 38, 95, 96]. Also, a well-known benchmark of 24 continuous constrained problems [89, 80] has been solved by PESO with different function evaluations (from 350,000 to 50,000) showing a robust performance [97].

A modified version of PESO, the PESDRO algorithm was developed to optimize binary systems [92]. Furthermore, a real-world problem concerning a behavior prediction in the Eurovision Song Contest 2008 was solved using the PESO algorithm [105].

Now, a modified version of the PESO algorithm is presented in the next section to solve the RVRP introduced in Chapter 4.

6.5 Modified PESO Algorithm for the RVRP

In some works, PSO algorithm has been used to solve the standard VRP [135, 84]. For solving the RVRP, this work uses a modified version of the PESO algorithm. A new partition strategy called *sectors model* and adapted perturbation operators (C-perturbation and M-perturbation) were developed to solve a combinatorial VRP optimization subject to reliability constraints.

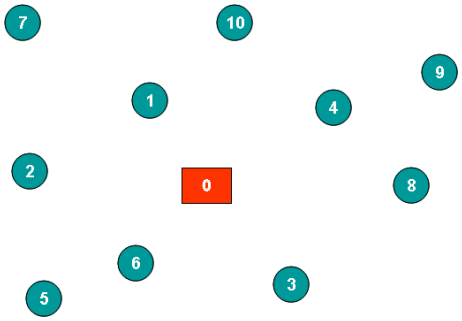


Figure 6.5: VRP nodes geographically scattered

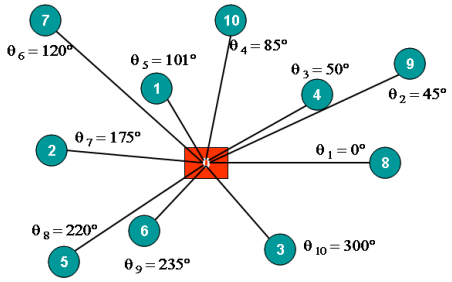


Figure 6.6: Customer position in polar coordinates

6.5.1 Sectors Model

The solution of a VRP instance (see Figure 6.5) is a set of least cost routes from one depot to a set of geographically scattered points. Every route is concerned with a customers subset located in the same neighborhood. A smart space partitioning of the customers set can save a lot of computational effort [24]. Spatial clustering can be applied to group similar spatial customers together; the implicit assumption is that patterns in space tend to be grouped rather than randomly located. A new spatial data mining technique for VRP, called **sectors model**, is proposed for clustering the customers according ito ts spatial localization.

First, the customers are sorted according to their polar angle θ, which is a geographical point in polar coordinates (r, θ) (see Fig. 6.6). Then, the customers and the depot can be inscribed by a circumference centered at D =depot, as shown in Fig. 6.7.

Several approaches perform a cluster applying a Euclidean distance. We propose to assign customers to vehicles by partitioning the customers into sectors according to their polar coordinates. The number of partitions (sectors) represents the number of vehicles allocated to find a VRP solution. The angles of sector limits (partition lines) are variables of the model. Figure 6.8 shows a partition of the customers into 3 sectors. The sectors can share a portion of

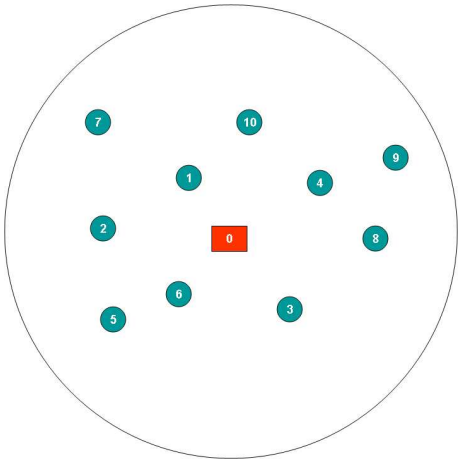


Figure 6.7: VRP Representation

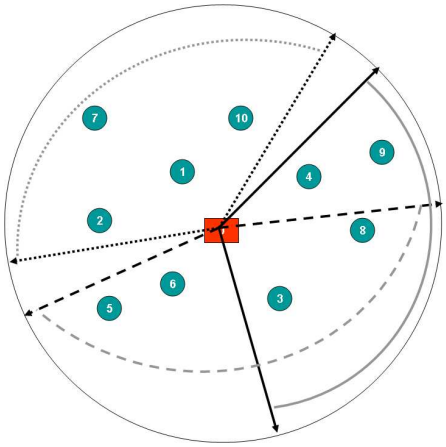


Figure 6.8: Customers partitioned into sectors

the customer set as it is shown in Fig. 6.8.

Customers within each sector are routed minimizing distance and waiting time. Customers into sharing regions are assigned to only one sector with probability $p = 1/s_i$, where s_i is the number of sectors sharing customer i. Figure 6.9 shows an instance of the possible customer assigment and routes obtained for the example shown in Fig. 6.8.

6.5.2 Representation

The sectors model (explained above) can be represented by a solution vector (particle) of dimension $d = 1 + 1.5 * l + n$

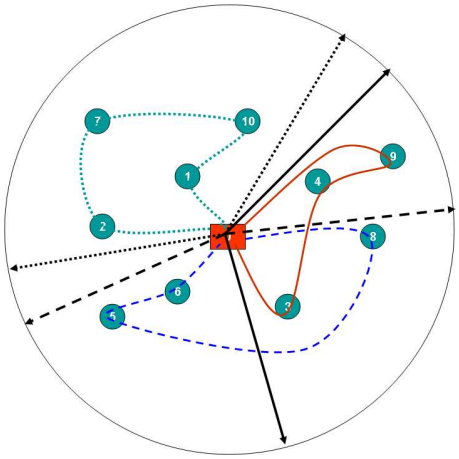


Figure 6.9: Routes design

$$\vec{x} = \{x_0, x_1, \ldots, x_d\}$$

where x_0 represents the number of sectors (vehicles), components x_1 to x_l represent the polar angle set ($l = 2 * x_0$), the components x_{s+1} to x_{l+n} are a permutation of the customer set $N = \{1, 2, \ldots, n\}$, and components x_{l+n+1} to $x_{l+n+l/2}$ represent the route size set. The route size determines the end and the begining of each route in the customer set.

For instance, the next vector represents the solution shown in Fig. 6.9

$$\vec{x} = \{\underbrace{3}_{cars}, \quad \underbrace{75, 190, 280, 60, 200, 15}_{polar\ angles}, \quad \underbrace{\overbrace{2, 7, 10, 1}^{route-1}, \overbrace{3, 4, 9}^{route-2}, \overbrace{8, 5, 6}^{route-3}}_{customers}, \quad \underbrace{4, 3, 3}_{route\ sizes} \}$$

6.5.3 Algorithm

Algorithm 5 presents the pseudocode of the modified PESO to solve RVRP optimization problems.

First, the number of partitions and the polar angle set are randomly generated. The number of partitions is bounded by $maxcars = customers$ in the upper limit, and $mincars = (totaldemand)/(vehiclecapacity)$ in the lower limit. The polar angle limits are $0 \leq \theta \leq 360$. The customers are assigned to each route based on the sector partition generated.

Algorithm 5 Modified PESO for RVRP

Generate randomly $\vec{x}$ from x_0 to x_l
$[\vec{y}, \vec{s}]$=Assign and count customers for sector set $\vec{x}$
$P_0 = \vec{x} \cup \vec{y} \cup \vec{s}$
Assess travel cost C_0 of P_0
Assess MSS reliabity R_0 of P_0
Initialize $P_{Best} = P_0$, $C_{Best} = C_0$ and $R_{Best} = R_0$
for $t = 1$ to $maxgenerations$ **do**
  $\vec{x}$ = Apply a PSO with SLR neighborhood to P_t from p_0 to p_l
  $[\vec{y}, \vec{s}]$=Reassign and count customers for sector set $\vec{x}$
  $P_{t+1} = \vec{x} \cup \vec{y} \cup \vec{s}$
  Assess travel cost C_{t+1} of P_{t+1}
  Assess MSS reliabity R_{t+1} of P_{t+1}
  $[P_{Best}, C_{Best}, R_{Best}]$ = *Feasibility tournament* $[P_{Best}, C_{Best}, R_{Best}]$ vs $[P_{t+1}, C_{t+1}, R_{t+1}]$
  if $U(0,1) < p_C$ **then**
    P_{Temp} = **C-Perturbation** (P_{Best})
    Assess travel cost C_{Temp} of P_{Temp}
    Assess MSS reliabity R_{Temp} of P_{Temp}
    $[P_{Best}, C_{Best}, R_{Best}]$ = *Feasibility tournament* $[P_{Best}, C_{Best}, R_{Best}]$ vs $[P_{Temp}, C_{Temp}, R_{Temp}]$
  end if
  if $U(0,1) < p_C$ **then**
    P_{Temp} = **M-Perturbation** (P_{Best})
    Assess travel cost C_{Temp} of P_{Temp}
    Assess MSS reliabity R_{Temp} of P_{Temp}
    $[P_{Best}, C_{Best}, R_{Best}]$ = *Feasibility tournament* $[P_{Best}, C_{Best}, R_{Best}]$ vs $[P_{Temp}, C_{Temp}, R_{Temp}]$
  end if
  $P_t = P_{t+1}$
end for

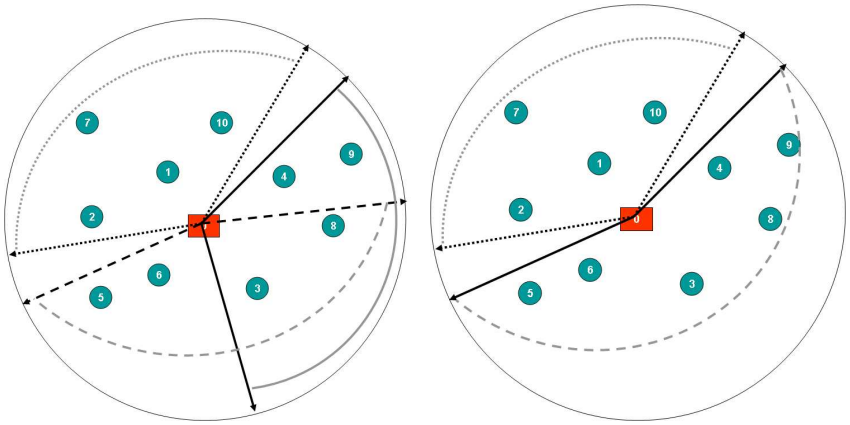


Figure 6.10: Two sectors joined after applying the Fusion route step

After initialization $P_{Best} = P_0$, a PSO with SLR neighborhood is applied. The step is only performed in the first $l + 1$ variables of each particle. The customers are reassigned to each route based on the sector partition generated by the PSO-SLR step. These variables compose the P_{t+1} population. A feasibility tournament is performed between P_{Best} and P_{t+1} to update P_{Best} population.

Next, the C-perturbation operator is performed on P_{Best}. Algorithm 6 shows the pseudocode of the adapted C-Perturbation operator for VRP.

Algorithm 6 Adapted C-perturbation operator for VRP

```
for k = 0 to swarmsize do
  for j = 0 to l do
    r = U(0, 1)
    p1 = Random(n)
    p2 = Random(n)
    PTemp[k, j] = PBest[k, j] + r (PBest[p1, j] - PBest[p2, j])
  end for
  PTemp[k,l+1:d] = Assign and count costumers for sector set PTemp[k,0 : l]
end for
Apply fusion route on PTemp
```

In the *fusion route* step (see Algorithm 6), two sectors are joined giving place to one larger sector that reduces the number of partitions (vehicles). The customers of one route are inserted in the other route. Figure 6.10 presents an example of a fusion route process.

Finally, the M-perturbation operator is performed on P_{Best}. This perturbation is implemented by adding small random numbers (from a uniform distribution) to the first $l + 1$ variables. Algorithm 7 shows the pseudocode of the adapted M-Perturbation operator for VRP.

Algorithm 7 Adapted M-perturbation operator for VRP

for $k = 0$ to $swarmsize$ **do**
  for $j = 0$ to l **do**
    $r = U(0, 1)$
    if $r \leq 1/l$ **then**
      $P_{Temp}[k, j] = \text{Rand}(LL, UL)$
    else
      $P_{Temp}[k, j] = P_{Best}[k, j]$
    end if
  end for
  $P_{Temp}[k,l+1{:}d]$ = Assign and count costumers for sector set $P_{Temp}[k,0 : l]$
  Apply *swapping customers* on $P_{Temp}[k,l+1{:}d]$
end for

In the *swapping customers* step (see Algorithm 7), two costumers into every sector are swapped with probability $1/customers$. It helps to find the optimal secuence for every route.

The PESO algorithm and its modified version to solve the RVRP are used in Chapter 7 to optimize the design MSS problems analyzed in this dissertation.

.

Chapter 7

Experiments

The experiments presented in this chapter are used to show the performance of the PESO algorithm detailed in Chapter 6. An artificial RRAP and a MSS network are used to compare PESO against other evolutionary algorithms applied in MSS reliability engineering. Besides, a real-world application of the RVRP introduced in Chapter 4 is presented to illustrate the RVRP theory.

7.1 Series-Parallel MSS: Reliability Allocation Problem

In this section, a reliability-redundancy allocation problem is used to compare our approach against two evolutionary algorithms. Tian et al. proposed a series-parallel MSS composed of three subsystems connected in series [127]. Every subsystem can posseses m_j items of three states $g_{j,i} = \{0.0, 0.5, 1.0\}$.

The objective optimization problem was formulated as follows [127]:

$$\textbf{Maximize} \qquad U = \sum_{k=1}^{3} g_k p_k \tag{7.1}$$

$$\textbf{S. t.} \qquad \sum_{j=1}^{3} c_j(p_{j,1}, p_{j,2}) \left[m_j + e^{\frac{m_j}{4}} \right] \leq C_0 \tag{7.2}$$

$$p_{j,i} \in [0,1], \qquad 0 \leq i \leq 2, 1 \leq j \leq 3 \tag{7.3}$$

$$\sum_{i=0}^{2} p_{j,i} = 1, \qquad 1 \leq j \leq 3 \tag{7.4}$$

$$0 < m_j, \qquad 1 \leq j \leq 3 \tag{7.5}$$

where C_0 is the maximum allowed system cost (budget) and U is the MSS utility, expected performance $E[G]$ (see Chapter 3). The cost of every item j, allocated into the parallel structure subsystems, as a function of its probabilities is

$$c_j(p_{j,1}, p_{j,2}) = \sum_{i=1}^{2} \alpha_{j,i} \left(\frac{-t}{ln(p_{j,i})} \right)^{\beta_{j,i}} \tag{7.6}$$

where $\alpha_{j,i}$ and $\beta_{j,i}$ are characteristic constants with respect to state i, and t is the operating time. Figure 7.1 shows the series-parallel MSS proposed by Tian et al. [127].

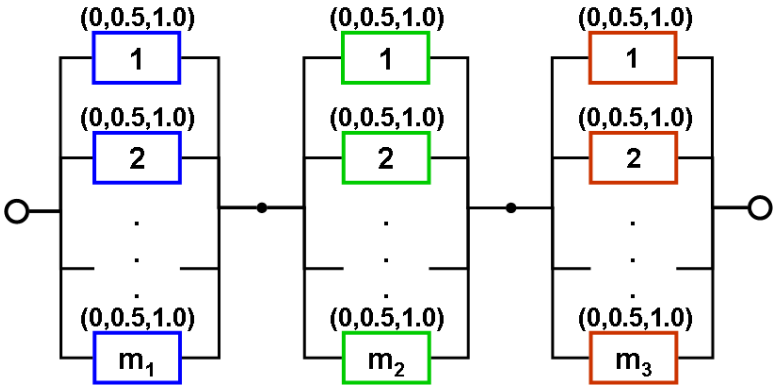


Figure 7.1: Series-parallel MSS: RRAP

The final system performance levels are $G = \{0, 0.5, 1.0\}$ [127]. The nine objective variables are n_j and $p_{j,i}$; $j = \{1, 2, 3\}$, $i = \{1, 2\}$. The cost function parameters $\alpha_{j,i}$ and $\beta_{j,i}$ are presented in Table 7.1. The system operating time is $t = 1000$.

Table 7.1: Cost function characteristic constants

Subsystem j	$p_{j,1}$	$p_{j,2}$	n_j	$\beta_{j,2}$
1	1.5E-05	4.0E-05	1.2	1.5
2	0.9E-05	3.2E-05	1.2	1.5
3	5.2E-05	9.0E-05	1.2	1.5

Tian et al. use a genetic algorithm to solve its proposed series-parallel MSS [127]. This problem was solved by Levitin et al. using a particle evolutionary swarm [77].

Table 7.2 presents the best results obtained by PESO, and reported by Tian et al. (GA)

[127] and Levitin et al. (PSO) [77]. The total system cost was set to $C_0 = 89.476$ for the comparison as it was reported by [77]. PESO uses a swarm size (population) of 100 particles and applies the PSO parameters detailed in Chapter 6.

Table 7.2: Comparision of the series-parallel MSS results

	Subsystem j	$\alpha_{j,1}$	$\alpha_{j,2}$	$\beta_{j,1}$	U	C
GA	1	0.2030	0.4200	8	0.9734	87.960264
	2	0.2109	0.4300	8		
	3	0.2100	0.4000	7		
PSO	1	0.2124	0.4579	7	0.9738	87.908320
	2	0.2208	0.4712	7		
	3	0.2042	0.4066	7		
PESO	1	0.262638	0.461075	7	0.97541	89.475952
	2	0.270919	0.474427	7		
	3	0.251862	0.409796	7		

Table 7.3 presents the results obtained by PESO over 30 runs and reported by Levitin et al. (PSO) over 200 runs [77]. For every run, 100,000 and 400,000 evaluations were performed by PESO and PSO, respectively.

Table 7.3: Comparision of PESO vs PSO

	Best	**Mean**	**Median**	**Worst**	**Std. Dev.**
PSO	0.9738	0.9734	-	0.9712	5.15E-04
PESO	0.975410	0.975390	0.975410	0.975302	3.6820E-05

PESO improves the best results reported by Tian et al. [127] and Levitin et al. [77]. Also, the worst value obtained by the PESO algorithm is better than the best solution obtained by these works. Note that PESO performs just 25% of the fitness evaluations applied by the PSO proposed by Levitin et al. [77].

7.2 Series-Parallel Binary System: Confidence Bounds

Figure 7.2 shows the series-parallel binary system proposed by Ramirez-Marquez and Jiang [113]. The system is a variation of the test system presented by Coit [13].

Table 7.4 provides the nominal reliability data for every binary item. The nominal system reliability is 0.852738.

For assessment the accuracy of the variance approximation introduced in Chapter 5, we apply the following experiment proposed by Coit [13] and also used by Ramirez-Marquez and

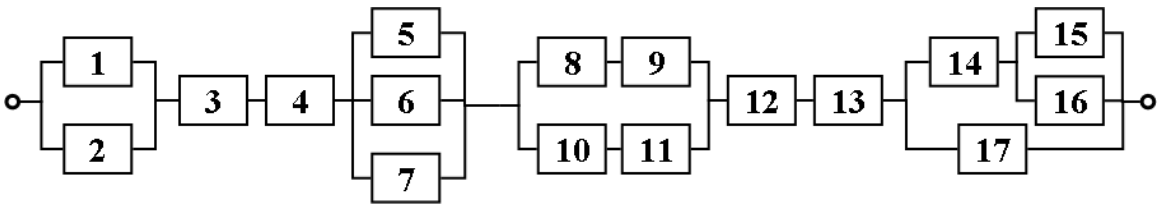


Figure 7.2: Series-parallel binary system

Table 7.4: Nominal reliabilities for the series-parallel binary system

Item	**Reliability**	**Item**	**Reliability**	**Item**	**Reliability**
1	0.91	7	0.80	13	0.99
2	0.91	8	0.92	14	0.90
3	0.96	9	0.95	15	0.83
4	0.98	10	0.92	16	0.83
5	0.80	11	0.95	17	0.88
6	0.80	12	0.96		

Jiang [113]. First, we generate binomial data associated with tha nominal probabilities shown in Table tab:nominal to simulate the behavior of the 17 items. 10 different data sample sizes are generate to test the robustness of our approach. Every sample has been constructed by starting the number of available testing units per component at 10 units, and then subsequently incrementing this number by 10, until a sample size of 100 units. Table 7.5 presents the results of a single test run when considering a data sample size of 50 testing units per component.

For each data sample, a confidence lower bound is constructed applying the technique introduced by Ramirez-Marquez and Jiang [113]. The distribution of the system reliability $R(t)$ is assumed unknown. The technique assumes that system reliability distribution can be well approximated by a Gaussian distribution. These assumptions allow for the computation of a $(1-\alpha)\%$ level confidence lower bound for $R(t)$:

$$\hat{R}(t) - Z_{1-\alpha}\sqrt{Var(R(t))} - \frac{1}{2N} \leq R(t) \tag{7.7}$$

Two α levels were suggested by Ramirez-Marquez and Jiang [113], $\alpha = 10\%$ and $\alpha = 5\%$. $100,000$ confidence lower bounds were constructed for every sample size $(10, \cdots, 100)$. Table 7.6 shows the coverage of the confidence lower bound for a desired $\alpha = 10\%$. The table presents the coverage results obtained by 3 approximation methods: Ramirez and Jiang [113], Delta1 (first order Taylor expansion) and Delta2 (second order Taylor expansion).

Table 7.7 shows the coverage of the confidence lower bound for a desired $\alpha = 5\%$ level.

Table 7.5: Binomial simulation for a sample size of 50 units

Item	**Failures**	$\hat{r}_i$	$Var(\hat{r}_i)$
1	4	0.92	0.001472
2	6	0.88	0.002112
3	1	0.98	0.000392
4	1	0.98	0.000392
5	8	0.84	0.002688
6	10	0.80	0.003200
7	8	0.84	0.002688
8	1	0.98	0.000392
9	3	0.94	0.001128
10	4	0.92	0.001472
11	1	0.98	0.000392
12	1	0.98	0.000392
13	0	1	0
14	5	0.90	0.001800
15	6	0.88	0.002112
16	7	0.86	0.002408
17	3	0.94	0.001128

Table 7.6: Coverage of the confidence lower bounds α=10%

Sample	**Ramirez and Jiang**		**Delta1**		**Delta2**	
size	**Coverage**	**Error**	**Coverage**	**Error**	**Coverage**	**Error**
10	22.37%	123.70%	21.04%	110.40%	18.45%	84.5%
20	10.42%	4.20%	10.33%	3.30%	10.09%	0.90%
30	11.75%	17.50%	11.65%	16.50%	11.40%	14.00%
40	10.65%	6.50%	10.60%	6.00%	10.36%	3.60%
50	10.84%	8.40%	10.77%	7.70%	10.55%	5.50%
60	10.79%	7.90%	10.76%	7.60%	10.68%	6.80%
70	10.32%	3.20%	10.26%	2.60%	10.09%	0.90%
80	10.54%	5.40%	10.46%	4.60%	10.33%	3.30%
90	10.33%	3.30%	10.28%	2.80%	10.12%	1.20%
100	10.38%	3.80%	10.32%	3.20%	10.28%	2.80%

Table 7.7: Coverage of the confidence lower bounds α=5%

Sample	Ramirez and Jiang		Delta1		Delta2	
size	Coverage	Error	Coverage	Error	Coverage	Error
10	19.47%	289.40%	17.77%	255.40%	14.26%	185.20%
20	9.18%	83.60%	9.04%	80.80%	8.81%	76.20%
30	8.60%	72.00%	8.41%	68.20%	7.91%	58.20%
40	7.06%	41.20%	6.96%	39.20%	6.67%	33.40%
50	6.67%	33.40%	6.62%	32.40%	6.44%	28.80%
60	6.69%	33.80%	6.62%	32.40%	6.43%	28.60%
70	6.52%	30.40%	6.44%	28.80%	6.25%	25.00%
80	6.35%	27.00%	6.34%	26.80%	6.25%	25.00%
90	6.20%	24.00%	6.17%	23.40%	6.04%	20.80%
100	6.08%	21.60%	6.04%	20.80%	5.95%	19.00%

The comparisons in Tables 7.6 and 7.7 show that the approximation based on Delta method with second order Taylor expansion provides the best coverage against the desired α level. As an example, the calculated α=5% for data sample size 50 is closer to a α=6.44% based on the simulation results. The coverage results in Tables 7.6 and 7.7 are all higher than the desired α levels. Nevertheless, as the sample size becomes larger, the simulation results shows that the coverage of the confidence lower bound becomes closer to the desired α.

7.3 Corbi Bakery: RVRP

In this section, the Corbi bakery located in Alcoy, Alicante, Spain is used as example. Corbi possess one bread plant and 18 bakeries (see Table 7.8). Figure 7.3 presents a urban map of the Alcoy city [145]. More information about the Corbi bakery case is available in [144].

Table 7.8: Corbi bakeries

ID	Bakery	ID	Bakery	ID	Bakery	ID	Bakery
0	Central	5	Carrascar	10	Espi	15	Isabel
1	Alamora	6	Cocentaina	11	Goncal	16	Isidre
2	Azorin	7	Eloi	12	Gregori	17	Poveda
3	Banyeres	8	Entenza28	13	Hispanitat	18	Torregrossa
4	Carbonell	9	Entenza99	14	Ibi		

Alcoy is known as *the city of the three rivers*, since it is divided by the Riquer, Benisaidó and Molinar rivers, tributaries of the Serpis river. Thereby, the bridges are the main urban characteristic of Alcoy, which have overcome the orographic irregularities of the town. *The city of bridges* possess some important festivals declared to be of International Tourist Interest. For

Figure 7.3: The Corbi bakery Case

example, the festival of Moors and Christians usually takes place on April, closing for 3 days some streets and avenues.

In a city as Alcoy, which possesses a limited link set (streets) to intercommunicate its zones, alternative routes to avoid possible interrupted traffic flows are essentials. The motivation to choose an example as the Corbi bakery comes from the weakness of its traffic network, the bridges. Figure 7.4 shows the optimal route (white line) for the Corbi bakery case for the standard VRP, which performs a travel distance of 14.2 units with 6 bridges in the route. An alternative route is not given at the event of a bridge failure (or other link), but a solution is developed using the proposed RVRP.

By applying the proposed RVRP, a set of alternative routes can be obtained to avoid bridge failures (interruptions). Two experiments were performed to show the impact of different reliability values in the RVRP solution.

In the experiments, only 3 out of 6 mentioned bridges are considered subject to failure. An additional bridge is included in the analysis to provide alternative routes in the RVRP. A probability of failure is assumed for every bridge shown in Figure 7.4 (blue arrows): A=0.9972,

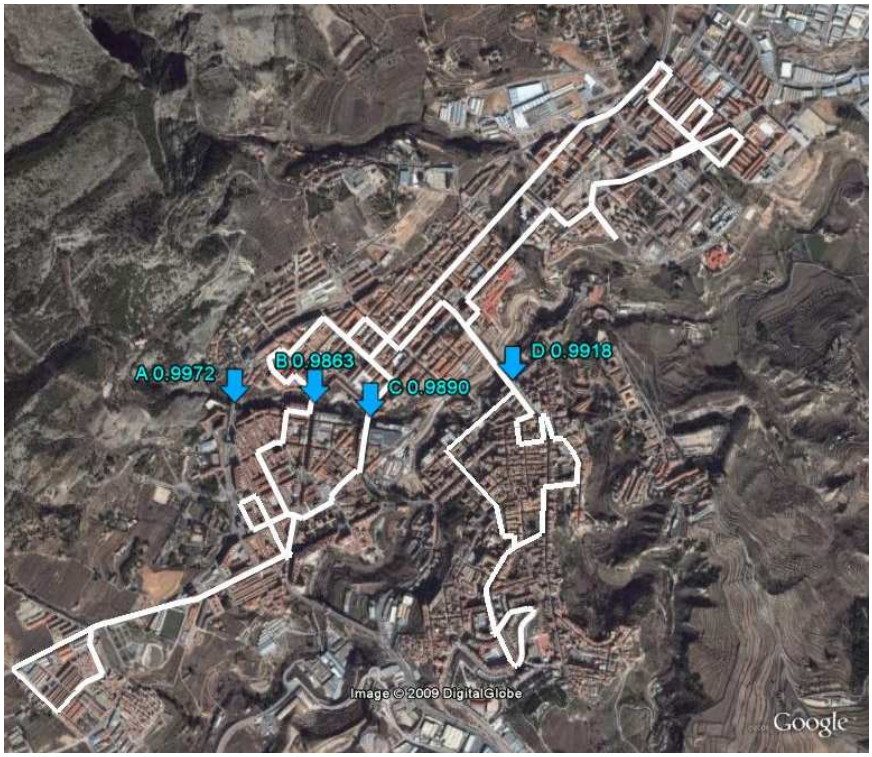


Figure 7.4: Corbi optimal route

B=0.9863, C=0.9890 and D=0.9918. The bridges B and C are one-way roads, and the bridges A and D are two-way roads. Alternative links for several pair of nodes (i, j) were found (see Figure 7.5).

In Figure 7.6, the Corbi's matrix C of distances and reliability factor is presented. It was approximated using the data base of Google Earth version 4.3 (related information about this software can be found in [145]). Every cell c_{ij} contains information of up to 3 links between nodes i and j, where each link $1 \leq k \leq 3$ has a data line. Each data line l_k shown in the cell c_{ij}, represents an alternative arc between the nodes i and j, with only 2 states: available and unavailable. The information provided in each data line l_k is the travel cost c_{ijk}, and its reliability p_{ijk}. The travel cost of an unavailable state of the link l_k is considered ∞, and its reliability $q_{ijk} = 1 - p_{ijk}$.

For example, the cell $(2, 0)$ contains two data lines: $(0.8, 0.9863)$ and $(1.2, 0.9972)$. Namely, there are two alternative paths between nodes 2 (Azorin) and 0 (Central), with travel cost of 0.8 and 1.2, and a reliability factor of 0.9863 and 0.9972, respectively. Their corresponding unreliability values are 0.0137 and 0.0028, for a travel cost of ∞ (interrupted path). The u-functions of the arcs l_k allocated in cell $(2, 0)$ are:

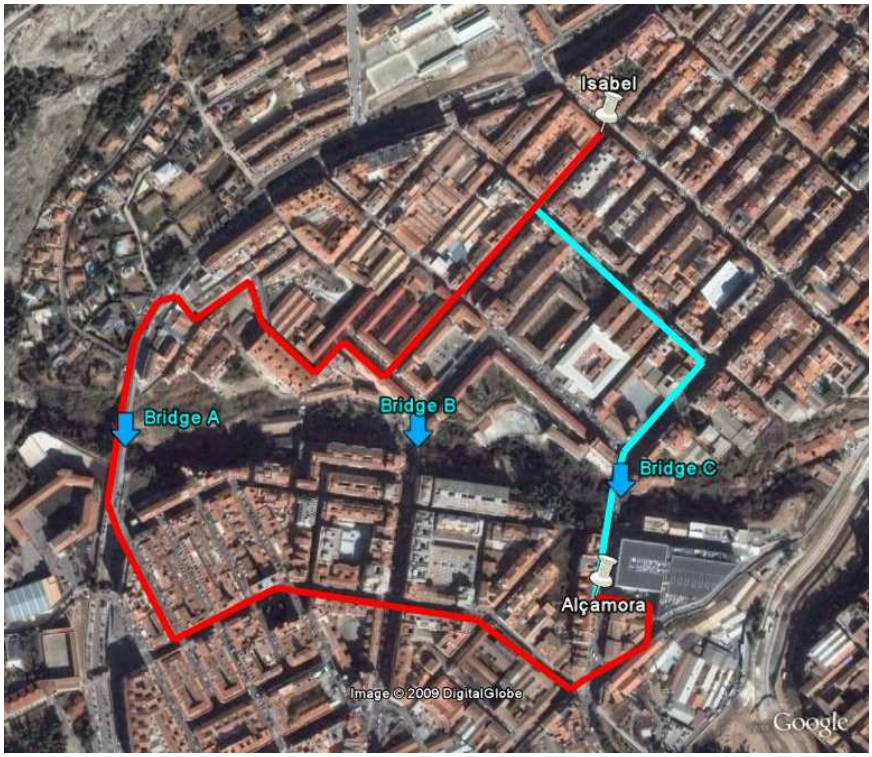


Figure 7.5: Alternative links for Alcamora-Isabel vertix

$$u_1(z) = 0.9863z^{0.8} + 0.0137z^{\infty}$$
$$u_2(z) = 0.9972z^{1.2} + 0.0028z^{\infty}$$

The u-function of the arc set to travel from Azorin to Central $(2, 0)$ is calculated as follows:

$$\begin{aligned}
U_{2,0}(z) &= \otimes_{min}(u_1(z), u_2(z)) \\
U_{2,0}(z) &= 0.9863(0.9972)z^{min(0.8,1.2)} + 0.9863(0.0028)z^{min(0.8,\infty)} \\
&\quad +0.0137(0.9972)z^{min(\infty,1.2)} + 0.0137(0.0028)z^{min(\infty,\infty)} \\
U_{2,0}(z) &= 0.983538z^{0.8} + 0.002762z^{0.8} + 0.013662z^{1.2} + 0.000038z^{\infty} \\
U_{2,0}(z) &= 0.9863z^{0.8} + 0.013662z^{1.2} + 0.000038z^{\infty}
\end{aligned}$$

In a similar way, the u-function $U_{i,j}(z)$ for every pair of nodes (i, j), $i \neq j \in 0, 1, \cdots, N$ is

ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0, 1.0	0.5, 1.0	1.2, 0.9972 1.3, 0.9890	1.0, 1.0	2.2, 1.0	1.4, 1.0	3.0, 0.9890 3.9, 0.9972	2.8, 0.9890 3.7, 0.9972	1.1, 0.9890 1.7, 0.9972	1.5, 0.9890 2.4, 0.9972	1.2, 1.0	2.3, 1.0	0.6, 1.0	2.4, 0.9890 3.4, 0.9972	0.6, 1.0	1.2, 0.9890 1.7, 0.9972	0.9, 0.9890 2.3, 0.9972	2.1, 0.9890 3.1, 0.9972	2.0, 0.9809 2.9, 0.9891 3.5, 1.0
1	1.0, 1.0	0, 1.0	0.7, 0.9890 1.3, 0.9972	2.0, 1.0	2.8, 0.9809 4.3, 0.9891	2.3, 1.0	2.5, 0.9890 4.0, 0.9972	2.3, 0.9890 3.8, 0.9972	1.2, 0.9890 1.8, 0.9972	1.0, 0.9890 2.5, 0.9972	0.7, 1.0	2.3, 0.9809 3.8, 0.9891	1.2, 1.0	1.9, 0.9890 3.5, 0.9972	1.0, 1.0	0.7, 0.9890 1.8, 0.9972	0.4, 0.9890 2.4, 0.9972	1.6, 0.9890 3.2, 0.9972	1.5, 0.9809 3.0, 0.9891
2	0.8, 0.9863 1.2, 0.9972	1.3, 0.9863 1.6, 0.9972	0, 1.0	1.6, 0.9972 1.8, 0.9863	3.2, 0.9918 3.4, 0.9972	1.9, 0.9972 2.2, 0.9863	3.0, 1.0	2.9, 1.0	2.4, 0.9972 3.0, 0.9890	1.4, 1.0	0.5, 0.9863 0.7, 0.9972	2.7, 0.9918 3.8, 0.9972	1.0, 0.9863 1.0, 0.9972	2.5, 1.0	0.8, 0.9863 0.8, 0.9972	0.7, 1.0	1.0, 1.0	2.2, 1.0	2.0, 0.9918 4.5, 0.9972
3	1.0, 1.0	2.5, 1.0	1.9, 0.9972 3.2, 0.9890	0, 1.0	2.9, 1.0	0.4, 1.0	4.6, 0.9972 5.0, 0.9890	4.4, 0.9972 4.8, 0.9890	2.4, 0.9972 3.0, 0.9890	3.1, 0.9972 3.5, 0.9890	1.7, 1.0	3.0, 1.0	0.8, 1.0	4.0, 0.9972 4.4, 0.9890	1.3, 1.0	2.3, 0.9972 3.1, 0.9890	2.7, 0.9972 2.9, 0.9890	3.7, 0.9972 4.1, 0.9890	4.0, 1.0
4	3.0, 1.0	2.1, 1.0	2.2, 0.9890 2.6, 0.9972	3.1, 1.0	0, 1.0	3.4, 1.0	3.6, 0.9918 3.9, 0.9890	3.5, 0.9918 3.8, 0.9890	2.0, 0.9890 2.4, 0.9918	2.2, 0.9918 2.5, 0.9890	2.7, 1.0	0.2, 1.0	2.7, 1.0	3.1, 0.9918 3.4, 0.9890	2.6, 1.0	2.1, 0.9890 2.3, 0.9918	1.9, 0.9890 2.3, 0.9918	2.8, 0.9918 3.1, 0.9890	1.2, 1.0
5	1.7, 1.0	3.2, 1.0	2.6, 0.9972 3.9, 0.9890	0.7, 1.0	3.6, 1.0	0, 1.0	5.3, 0.9972 5.7, 0.9890	5.1, 0.9972 5.5, 0.9890	3.1, 0.9972 3.7, 0.9890	3.8, 0.9972 4.2, 0.9890	2.4, 1.0	3.7, 1.0	1.5, 1.0	4.7, 0.9972 5.1, 0.9890	2.0, 1.0	3.0, 0.9972 3.8, 0.9890	3.4, 0.9972 3.6, 0.9890	4.4, 0.9972 4.8, 0.9890	4.7, 1.0
6	2.7, 0.9863 3.7, 0.9972	3.4, 0.9863 4.1, 0.9972	2.5, 1.0	3.8, 0.9863 4.1, 0.9972	3.9, 0.9918 5.6, 0.9863	4.1, 0.9863 4.4, 0.9972	0, 1.0	0.2, 1.0	1.8, 1.0	1.4, 1.0	2.4, 0.9863 3.2, 0.9972	3.4, 0.9918 5.1, 0.9863	3.0, 0.9863 3.5, 0.9972	0.2, 1.0	2.8, 0.9863 3.3, 0.9972	2.2, 1.0	2.4, 1.0	0.6, 1.0	2.6, 0.9918 6.3, 0.9863
7	2.6, 0.9863 3.6, 0.9972	3.3, 0.9863 4.0, 0.9972	2.4, 1.0	3.7, 0.9863 4.0, 0.9972	3.8, 0.9918 5.5, 0.9863	4.0, 0.9863 4.3, 0.9972	0.6, 1.0	0, 1.0	1.7, 1.0	1.3, 1.0	2.3, 0.9863 3.1, 0.9972	3.3, 0.9918 5.0, 0.9863	2.9, 0.9863 3.4, 0.9972	0.4, 1.0	2.7, 0.9863 3.2, 0.9972	2.1, 1.0	2.4, 1.0	1.0, 1.0	2.5, 0.9918 6.2, 0.9863
8	0.9, 0.9863 1.8, 0.9972	1.4, 0.9863 2.2, 0.9972	0.7, 1.0	1.9, 0.9863 2.2, 0.9972	2.4, 0.9918 4.4, 0.9972	2.3, 0.9863 2.6, 0.9972	2.2, 1.0	1.8, 1.0	0, 1.0	0.6, 1.0	0.6, 0.9863 1.3, 0.9972	1.9, 0.9918 4.0, 0.9972	1.1, 0.9863 1.6, 0.9972	1.6, 1.0	0.9, 0.9863 1.4, 0.9972	0.3, 1.0	0.6, 1.0	1.3, 1.0	1.2, 0.9918 5.1, 0.9972
9	1.4, 0.9863 2.3, 0.9972	2.0, 0.9863 2.7, 0.9972	1.1, 1.0	2.4, 0.9863 2.7, 0.9972	2.4, 0.9918 4.9, 0.9972	2.7, 0.9863 3.1, 0.9972	2.1, 1.0	2.0, 1.0	0.5, 1.0	0, 1.0	1.1, 0.9863 1.8, 0.9972	1.9, 0.9918 4.5, 0.9972	1.6, 0.9863 2.1, 0.9972	1.6, 1.0	1.4, 0.9863 1.9, 0.9972	0.8, 1.0	1.1, 1.0	1.3, 1.0	1.2, 0.9918 5.6, 0.9972
10	0.5, 1.0	0.8, 1.0	0.8, 0.9972 1.0, 0.9890	1.5, 1.0	3.1, 0.9809 3.2, 1.0	1.8, 1.0	2.7, 0.9890 3.5, 0.9972	2.5, 0.9890 3.3, 0.9972	0.8, 0.9890 1.4, 0.9972	1.2, 0.9890 2.0, 0.9972	0, 1.0	2.6, 0.9809 2.8, 1.0	0.6, 1.0	2.2, 0.9890 3.0, 0.9972	0.3, 1.0	0.9, 0.9890 1.2, 0.9972	0.7, 0.9890 1.9, 0.9972	1.9, 0.9890 2.7, 0.9972	1.7, 0.9809 2.6, 0.9891 4.0, 1.0
11	1.7, 1.0	1.3, 1.0	2.0, 0.9890 2.4, 0.9972	2.3, 1.0	0.5, 1.0	2.6, 1.0	3.5, 0.9918 5.1, 0.9972	3.4, 0.9918 5.0, 0.9972	1.8, 0.9890 2.3, 0.9918 3.0, 0.9972	2.3, 0.9890 3.1, 0.9918 3.6, 0.9972	2.0, 1.0	0, 1.0	1.9, 1.0	3.0, 0.9918 4.6, 0.9972	1.9, 1.0	2.0, 0.9890 2.2, 0.9918 2.9, 0.9972	1.7, 0.9890 2.3, 0.9918 3.4, 0.9972	2.7, 0.9918 4.3, 0.9972	1.1, 1.0
12	0.2, 1.0	0.7, 1.0	0.9, 0.9972 1.4, 0.9890	0.9, 1.0	2.2, 1.0	1.3, 1.0	3.1, 0.9890 3.6, 0.9972	2.9, 0.9890 3.4, 0.9972	1.2, 0.9890 1.5, 0.9972	1.7, 0.9890 2.1, 0.9972	0.7, 1.0	2.4, 1.0	0, 1.0	2.5, 0.9890 3.1, 0.9972	0.3, 1.0	1.3, 0.9890 1.4, 0.9972	1.1, 0.9890 2.0, 0.9972	2.2, 0.9890 2.8, 0.9972	2.1, 0.9809 2.7, 0.9891 3.5, 1.0
13	2.5, 0.9863 3.5, 0.9972	3.0, 0.9863 3.9, 0.9972	2.2, 1.0	3.5, 0.9863 3.9, 0.9972	3.6, 0.9918 6.1, 0.9972	3.9, 0.9863 4.3, 0.9972	0.5, 1.0	0.2, 1.0	1.6, 1.0	1.1, 1.0	2.3, 0.9863 3.0, 0.9972	3.1, 0.9918 5.7, 0.9972	2.8, 0.9863 3.3, 0.9972	0, 1.0	2.6, 0.9863 3.1, 0.9972	1.9, 1.0	2.2, 1.0	0.6, 1.0	2.3, 0.9918 6.8, 0.9972
14	0.2, 1.0	0.7, 1.0	0.8, 0.9972 1.4, 0.9890	1.3, 1.0	2.4, 1.0	1.6, 1.0	3.1, 0.9890 3.6, 0.9972	2.9, 0.9890 3.4, 0.9972	1.2, 0.9890 1.4, 0.9972	1.7, 0.9890 2.1, 0.9972	0.4, 1.0	2.5, 1.0	0.3, 1.0	2.5, 0.9890 3.0, 0.9972	0, 1.0	1.4, 0.9890 1.3, 0.9972	1.1, 0.9890 1.9, 0.9972	2.2, 0.9890 2.7, 0.9972	2.1, 0.9809 2.6, 0.9891 3.7, 1.0
15	0.9, 0.9863 1.7, 0.9972	1.4, 0.9863 2.1, 0.9972	0.7, 1.0	1.8, 0.9863 2.1, 0.9972	2.7, 0.9918 4.3, 0.9972	2.2, 0.9863 2.4, 0.9972	2.3, 1.0	2.1, 1.0	0.1, 1.0	0.7, 1.0	0.6, 0.9863 1.2, 0.9972	2.2, 0.9918 3.9, 0.9972	1.1, 0.9863 1.5, 0.9972	1.8, 1.0	0.9, 0.9863 1.3, 0.9972	0, 1.0	0.6, 1.0	1.5, 1.0	1.3, 0.9918 5.0, 0.9972
16	0.8, 0.9863 1.8, 0.9972	1.3, 0.9863 2.2, 0.9972	0.6, 1.0	1.8, 0.9863 2.2, 0.9972	2.9, 0.9918 4.4, 0.9972	2.2, 0.9863 2.5, 0.9972	2.6, 1.0	2.4, 1.0	0.4, 1.0	1.1, 1.0	0.5, 0.9863 1.3, 0.9972	2.4, 0.9918 4.0, 0.9972	1.1, 0.9863 1.6, 0.9972	2.0, 1.0	0.8, 0.9863 1.4, 0.9972	0, 1.0	0, 1.0	1.7, 1.0	1.6, 0.9918 5.1, 0.9972
17	2.4, 0.9863 3.3, 0.9972	2.8, 0.9863 3.7, 0.9972	2.1, 1.0	3.4, 0.9863 3.7, 0.9972	3.5, 0.9918 5.9, 0.9972	3.7, 0.9863 4.0, 0.9972	0.9, 1.0	0.8, 1.0	1.5, 1.0	1.0, 1.0	2.1, 0.9863 2.8, 0.9972	3.0, 0.9918 5.5, 0.9972	2.6, 0.9863 3.1, 0.9972	0.7, 1.0	2.4, 0.9863 2.9, 0.9972	1.8, 1.0	2.1, 1.0	0, 1.0	2.1, 0.9918 6.6, 0.9972
18	1.6, 1.0	1.2, 1.0	1.8, 0.9918 2.3, 0.9972	2.2, 1.0	1.8, 1.0	2.6, 1.0	2.4, 0.9918 5.0, 0.9972	2.2, 0.9918 4.8, 0.9972	1.1, 0.9918 2.9, 0.9972	0.9, 0.9918 3.5, 0.9972	1.8, 0.9782 2.4, 1.0	1.4, 1.0	1.8, 1.0	1.9, 0.9918 4.5, 0.9972	1.8, 1.0	1.1, 0.9918 2.8, 0.9972	1.1, 0.9918 3.4, 0.9972	1.5, 0.9918 4.2, 0.9972	0, 1.0

Figure 7.6: Matrix of travel costs and reliabilities for the Corbi bakery case

calculated. Two reliability values are used in the experiments: 0.98 and 0.999. They correspond to 357.70 and 364.635 available days during a year. The RVRP solution includes the total travel cost c_K, reliability p_K, and links for each route (visitation order). Besides, the reliability can be translated into available days in one year period of time A_k.

The modified version of PESO for solving the RVRP is applied with a population size of 100 particles and the PSO parameters mentioned in Chapter 6. PESO applies 350,000 fitness evaluations for every run. The results of the 30 runs performed by PESO constantly converge to the optimal value.

7.3.1 Corbi Bakery Case $R \geq 0.98$

Table 7.9 shows the RVRP solution of the Corbi bakery case subject to a total reliability of $R \geq 0.98$. The RVRP solution of the optimal route shown in Figure 7.4 has $K \in \{1, 2, \cdots, 9\}$ alternatives (states). In Table 7.9, the *Route K* column defines the visitation order of the route, where the subscript in every node denotes the arc $a_{i,j,k}$ used by the route. For example, in the first RVRP state ($K = 1$), the route begins with a path between nodes 0 and 3. The subscript in node 0 (a number 1) indicates that the arc number 1 is included in the route. This is represented as $a_{0,3,1}$, which has a travel distance of $c_{0,3,1} = 1.0$, and a reliability of $p_{0,3,1} = 1.0$ (see Figure 7.5).

Table 7.9: RVRP solution of Corbi bakery case ($R \geq 0.98$)

K	c_K	p_K	A_K	**Route K**
1	14.2	0.9595	350	$0_1, 3_1, 5_1, 12_1, 1_1, 16_1, 15_1, 8_1, 11_1, 4_1, 18_1, 17_1, 6_1, 13_1, 7_1, 9_1, 2_1, 10_1, 14_1$
2	14.4	0.0132	5	$0_1, 3_1, 5_1, 12_1, 1_1, 16_1, 15_1, 8_1, 11_1, 4_1, 18_1, 17_1, 6_1, 13_1, 7_1, 9_1, 2_2, 10_1, 14_1$
3	16.2	0.0106	4	$0_1, 3_1, 5_1, 12_1, 1_2, 16_1, 15_1, 8_1, 11_1, 4_1, 18_1, 17_1, 6_1, 13_1, 7_1, 9_1, 2_1, 10_1, 14_1$
4	16.4	1.47E-04	0	$0_1, 3_1, 5_1, 12_1, 1_2, 16_1, 15_1, 8_1, 11_1, 4_1, 18_1, 17_1, 6_1, 13_1, 7_1, 9_1, 2_2, 10_1, 14_1$
5	19.0	6.52E-05	0	$0_1, 3_1, 5_1, 12_1, 1_1, 16_1, 15_1, 8_2, 11_1, 4_1, 18_2, 17_1, 6_1, 13_1, 7_1, 9_1, 2_1, 10_1, 14_1$
6	19.2	9.03E-07	0	$0_1, 3_1, 5_1, 12_1, 1_1, 16_1, 15_1, 8_2, 11_1, 4_1, 18_2, 17_1, 6_1, 13_1, 7_1, 9_1, 2_2, 10_1, 14_1$
7	21.0	7.23E-07	0	$0_1, 3_1, 5_1, 12_1, 1_2, 16_1, 15_1, 8_2, 11_1, 4_1, 18_2, 17_1, 6_1, 13_1, 7_1, 9_1, 2_1, 10_1, 14_1$
8	21.2	1.00E-08	0	$0_1, 3_1, 5_1, 12_1, 1_2, 16_1, 15_1, 8_2, 11_1, 4_1, 18_2, 17_1, 6_1, 13_1, 7_1, 9_1, 2_2, 10_1, 14_1$
9	∞	0.0163	6	Total Failure

Figure 7.7 shows the state $K = 1$ of the optimal RVRP subject to $R \geq 0.98$. Note that the first RVRP solution, $K = 1$, represents the standard VRP solution. This route could be used only ≈ 350 out of 365 days. Namely, the standard VRP solution could be temporally unavailable ≈ 15 days during a year. Thereby, alternative routes could be used to avoid interrupted paths. For instance, the travel cost of the route increases to 14.4 units whenever bridge B fails (state $K = 2$). This alternative is required with probability $p = 0.0132$, that is approximately 5 days

during a year.

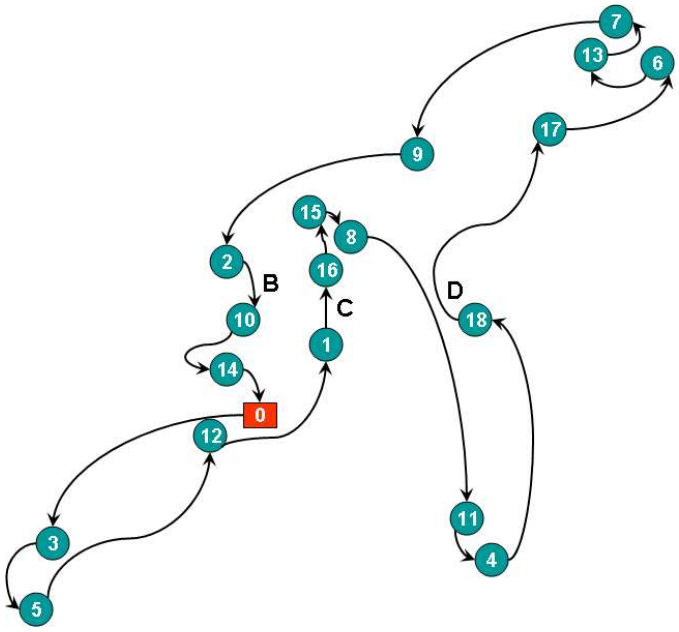


Figure 7.7: RVRP optimal route ($R \geq 0.98$), state $K = 1$

The route $K = 2$ replaces the unavailable arc $a_{2,10,1}$ (between nodes 2 and 10) with travel distance $c_{2,10,2} = 0.5$, and reliability $p_{2,10,2} = 0.9863$, with the alternative arc $a_{2,10,2}$ with travel distance $c_{2,10,2} = 0.7$ and reliability $p_{2,10,2} = 0.9972$. Figure 7.8 shows the alternative route when an interruption of traffic flow occurs at bridge B (state $K = 2$).

Similarly, Figure 7.9 illustrates the alternative route when an interruption of traffic flow occurs at bridge C (state $K = 3$). This alternative route, $K = 3$, replaces the unavailable arc $a_{1,16,1}$ with the alternative arc $a_{1,16,2}$.

The RVRP solution presented in Table 7.9 reports that there is not an alternative route to solve the interruption in the traffic network (state $K = 9$), approximately 6 days out of 365 days as it is shown in Figure 7.10. In the RVRP optimization, the reliability constraint is introduced as a system demand to design an optimal route available at least $R\%$ for a stated period of time. For this experiment, there are 8 available routes $K \in \{1, 2, \ldots, 8\}$, with a total reliability of $R = 0.9837$, that is ≈ 359 days during a year.

For designing a route with more available days than the current one, the reliability constraint R is increased in the next experiment.

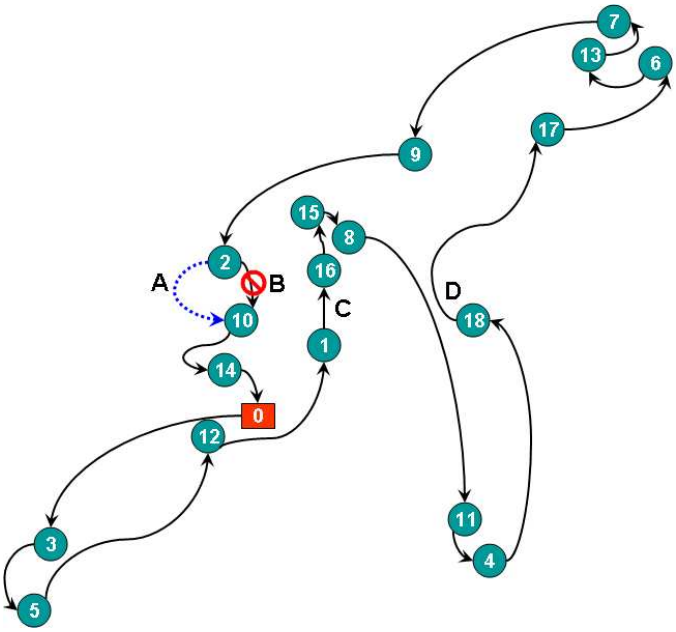


Figure 7.8: RVRP optimal route ($R \geq 0.98$), state $K = 2$

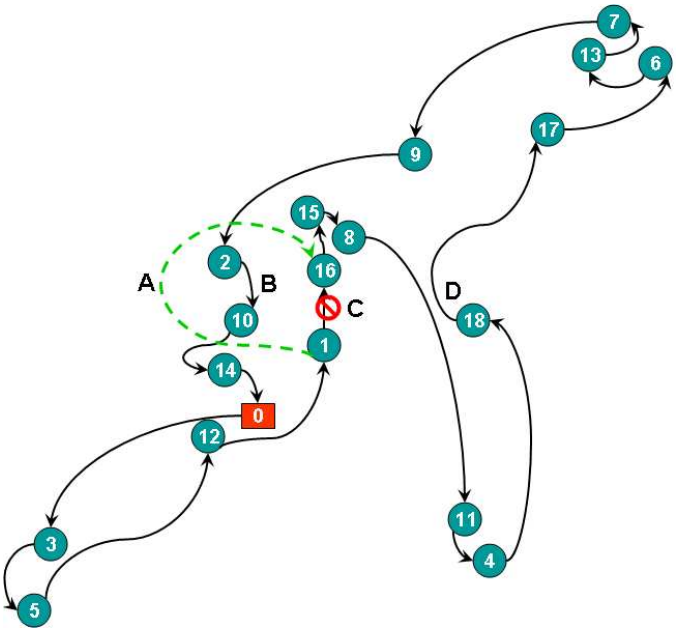


Figure 7.9: RVRP optimal route ($R \geq 0.98$), state $K = 3$

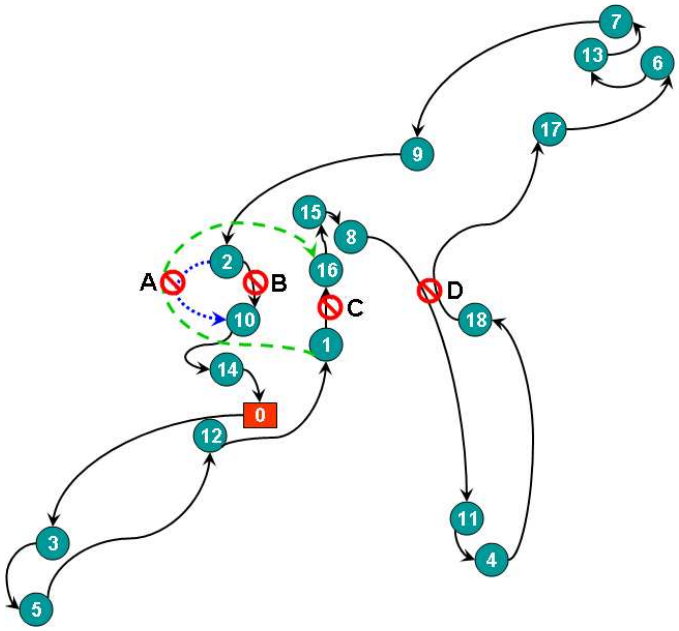


Figure 7.10: RVRP optimal route ($R \geq 0.98$), total failure state $K = 9$

7.3.2 Corbi Bakery Case $R \geq 0.999$

Table 7.10 shows the RVRP solution of the Corbi bakery case subject to a total reliability of $R \geq 0.999$ (≈ 365 days).

Table 7.10: RVRP solution of Corbi bakery case ($R \geq 0.999$)

K	c_K	p_K	A_K	**Route K**
1	14.8	0.9755	356	$0_1, 3_1, 5_1, 12_1, 4_1, 11_1, 18_1, 1_1, 17_1, 6_1, 13_1, 7_1, 9_1, 8_1, 15_1, 16_1, 2_1, 10_1, 14_1$
2	15.0	0.0135	5	$0_1, 3_1, 5_1, 12_1, 4_1, 11_1, 18_1, 1_1, 17_1, 6_1, 13_1, 7_1, 9_1, 8_1, 15_1, 16_1, 2_2, 10_1, 14_1$
3	16.4	0.0108	4	$0_1, 3_1, 5_1, 12_1, 4_1, 11_1, 18_1, 1_2, 17_1, 6_1, 13_1, 7_1, 9_1, 8_1, 15_1, 16_1, 2_1, 10_1, 14_1$
4	16.6	1.50E-04	0	$0_1, 3_1, 5_1, 12_1, 4_1, 11_1, 18_1, 1_2, 17_1, 6_1, 13_1, 7_1, 9_1, 8_1, 15_1, 16_1, 2_2, 10_1, 14_1$
5	∞	6.92E-05	0	Total Failure

The first RVRP solution (state $K = 1$) presented in Table 7.10 performs a travel distance higher than the standard VRP solution. But, this RVRP solution reports that a total failure state ($K = 5$) occurs with probability $p_\infty = 6.92E - 05$, that is, ≈ 0 days. Therefore, there is an alternative route always available to accomplish the customer demands. Figure 7.11 shows the first route (state $K = 1$) of the Corbi bakery case subject to a total reliability of $R \geq 0.999$, which is so different to the visitation sequence shown in Figure 7.7.

Figures 7.12 and 7.13 show the alternative routes when a failure apears in bridge B and C, respectively.

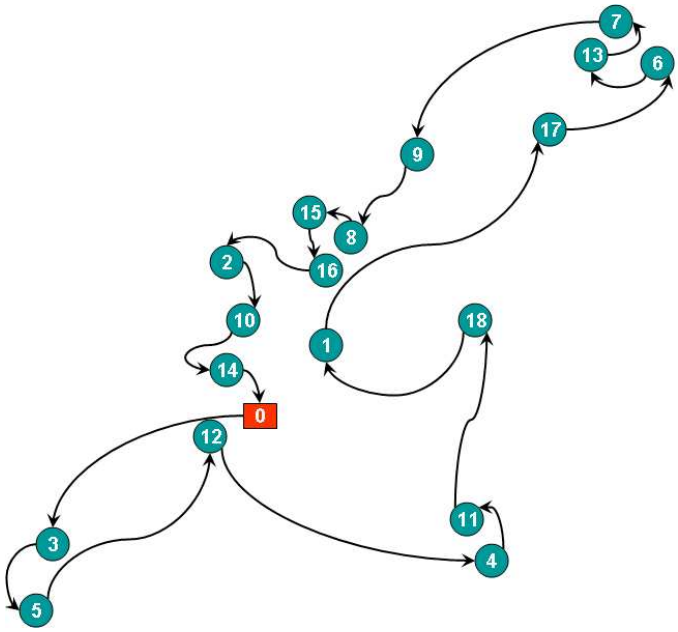


Figure 7.11: RVRP optimal route ($R \geq 0.999$), state $K = 1$

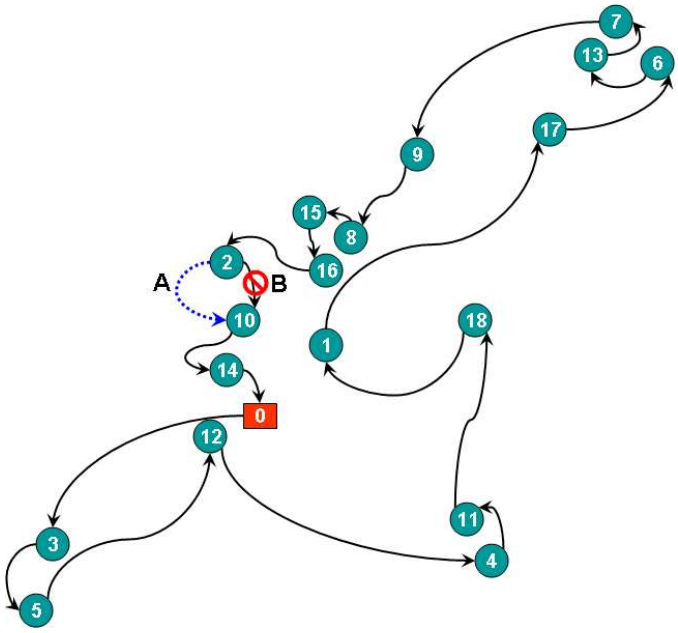


Figure 7.12: RVRP optimal route ($R \geq 0.999$), state $K = 2$

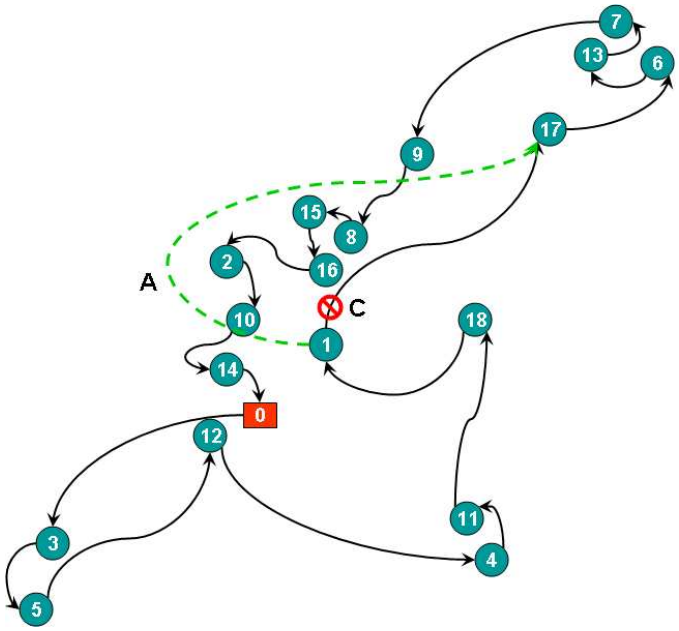


Figure 7.13: RVRP optimal route ($R \geq 0.999$), state $K = 3$

Observing the two experiments, it is clear that there is a tradeoff between availability and travel cost. As such, all transportation problems which intend to minimize total travel cost, are subject to the physical considerations of unavailable traffic paths or roadways. Thereby, an important optimization problem is part of the RVRP. The example presented in this section shows that it is possible to find a VRP solution with standing availability, impacting slightly on the travel cost.

7.3.3 Variance Approximation

For showing the Delta-UGF method described in Chapter 5, we assume the variance values presented in Table 7.11 for every bridge:

Table 7.11: Variance reliabilities for Corbi's bridges

Bridge	p_b	σ_b^2
A	0.9972	4.6913E-07
B	0.9863	5.7469E-06
C	0.9890	4.2222E-06
D	0.9918	1.8765E-06

Based on the Corbi bakery case subject to a total reliability of $R \geq 0.999$, we approximate the variance and covariance matrix of the RVRP solution shown in Table 7.10. Table 7.12 shows

the approximation of the variance matrix of the RVRP solution subject to $R \geq 0.999$.

Table 7.12: Variance matrix of the RVRP solution ($R \geq 0.999$)

Bridge	p_b	σ_b^2		
9.7284E-06	-5.5485E-06	-4.0335E-06	-1.1921E-07	-2.7240E-08
-5.5485E-06	5.5906E-06	-1.1881E-07	6.1359E-08	1.5377E-08
-4.0335E-06	-1.1881E-07	4.0851E-06	5.6042E-08	1.1167E-08
-1.1921E-07	6.1359E-08	5.6042E-08	1.5037E-09	3.3200E-10
-2.7240E-08	1.5377E-08	1.1167E-08	3.3200E-10	3.6357E-10

For building a confidence interval, we use the confidence lower bound applied in Section 7.2. The availability of the RVRP solution is the sum of the routes with a travel cost different to infinite ($c_K \neq \infty$).

$$\hat{R}(\hat{p}_1, \hat{p}_2, \hat{p}_3, \hat{p}_4) = \sum_{K=1}^{4} \hat{p}_K = 0.99995 \tag{7.8}$$

Applying the Delta method, we can approximate the variance of the function $\hat{R}(\hat{p}_1, \hat{p}_2, \hat{p}_3, \hat{p}_4)$.

$$Var(\hat{R}(\hat{p}_1, \hat{p}_2, \hat{p}_3, \hat{p}_4)) \approx \sum_{K=1}^{4} \sigma_{\hat{p}_K}^2 + 2 \sum_{K=1}^{3} \sum_{H=K+1}^{4} Cov\hat{p}_K, \hat{p}_H \tag{7.9}$$

$$Var(\hat{R}(\hat{p}_1, \hat{p}_2, \hat{p}_3, \hat{p}_4)) \approx 4.11958E - 10 \tag{7.10}$$

The computation of a $(1-\alpha)\%$ level confidence lower bound for $\hat{R}(\hat{p}_1, \hat{p}_2, \hat{p}_3, \hat{p}_4)$ is:

$$\begin{aligned} \hat{R} - Z_{1-\alpha}\sqrt{Var(R)} &\leq R \\ 0.99995 - Z_{0.95}\sqrt{4.11958E-10} &\leq R \\ 0.9999166 &\leq R \end{aligned}$$

The confidence lower bound $0.9999166 \leq R$ indicates that the route is available at least $364.9696 \approx 365$ days. Namely, the RVRP solution subject to $R \geq 0.999$ is available during 365 days with a confidence of 95%. This example provides a overview of the present dissertation: RVRP (modelation via UGF), variance approximation (via Delta-UGF method) and MSS reliability optimization (via PESO).

.

.

Chapter 8

Final Remarks

This chapter summarizes the work presented in this thesis. Posteriorly, a brief discussion of several future works is provided.

8.1 Summary

This thesis established as its objective to study the reliability of multi-state systems. The contributions are three:

1. **Approximating the MSS reliability variances**.
2. **Introducing the MSS reliability theory in the Vehicle Routing Problem**.
3. **Developing a evolutionary algorithm for the optimization of MSS relibility designs**.

8.1.1 MSS reliability variances

Nowadays, modern systems and equipment are characterized by complexity of structures and variety of initial information. In practice, the probability of item states are obtained on the basis of limited data so that they are affected by significant uncertainty [88].

Several methods for the assessment of MSS reliability are available in the literature. However, theses methods assume that the item reliabilities are accurate. Thereby, a practical approach for analyzing uncertainty in MSS reliability, can provide trustworthy designs and improve the engineering design strategies.

Chapter 5 describes our approach based on Delta method for the propagation of uncertainty in a MSS. The Delta method proposed in this thesis, uses second order Taylor series for approximating the reliability variances of the entire MSS. A computational method to approximate the MSS reliability variances was developed based on a combination of UGF-Delta method.

Approximating the variance of the entire MSS reliability can help to estimate the real expected performance of a MSS composed of items with uncertain probabilities.

8.1.2 MSS reliability in the VRP

Nowadays, efficient delivery transport systems are required for increasing the competitive power of industry. Even a tiny improvement in the efficiency in the vehicle routing process is transformed in a sensible monetary gain, due to the fact that the distribution process is repeated every day of the year and gains are easily cumulated over time [117].

The VRP have been studied since 1959, when it was introduced by Dantzig and Ramser [17]. From then on, many variants have been analized: the vehicle routing problem with time windows (VRPTW), the capacitated vehicle routing problem (CVRP), the multi-depot vehicle routing problem (MDVRP), the site-dependent vehicle routing problem (SDVRP), etc., [110]. Nevertheless, current works in the field of operational research have not studied the VRP with failures into the traffic network.

The assumption that in a real-world traffic network everything goes according to an *a priori* determined static schedule is unrealistic. In fact, in real-world traffic networks the availability of roadways or paths depends on traffic congestion, weather conditions, car accidents, roadworks, holidays, etc. Since the cost impact due to nonfulfillment delivering can be substantial, finding an optimal routing strategy under uncertainty is an important VRP variant, especially in emergency response and disaster management [25].

This dissertation applies the MSS reliability theory (availability) in the VRP. The key issue considered in Chapter 4 is to incorporate alternative traffic paths between two nodes communicated by a roadway likely to failures (unavailable).

The RVRP avoids unavailable traffic paths, using a set of alternative ways that possess an associated set of reliability values. This allows the accomplish of the entire route, sometimes with a slightly longer travel cost.

8.1.3 MSS reliability design via PESO algorithm

Many evolutionary algorithms have been used to solve reliability design problems. For example, Levitin and Lisnianski have proposed a series of methods on using the UGF approach and GA for the optimization of multi-state systems [83]. Although, GA is very popular in engineering design problems, other evolutionary algorithms can be applied, like ant colony optimization [104] or particle swarm optimization [77]. Since it has been proved that there is no a meta-heuristic that could always outperform other meta-heuristics in solving all kinds of problems [137]; introducing new optimization approaches can provide alternative algorithms able to solve different kind of problems or to outperform the current methods.

The PESO algorithm is a modified PSO that incorporates two perturbation operators to keep up population diversity. Besides, PESO applies a new PSO structure called singly-linked ring to form the local neighborhood of every particle, eliminating the mutual attraction between two neighbors. A constraint handling mechanism based on the `superiority of feasible solutions` proposed by Deb [18] is incorporated in PESO to handle equality and inequality constraints [38].

The performance of the PESO algorithm has been shown very robust in artificial and real-world problems with continuous and discrete variables [93, 94, 95, 96, 105]. Also, note the first version of PESO was applied to optimize the reliability design of binary systems [92].

PESO improves the results obtained by other evolutionary algorithms in the series parallel MSS presented in Chapter 7. Also, PESO finds the optimal results for the RVRP presented in Chapter 7. PESO shows that is able to solve different MSS design reliability problems.

8.2 Future Research

MSS reliability is a field with a promising future research. Nowadays, quality is a word that sounds loudly in manufacture and service companies. As it was mentioned in the introduction of this work, reliability is concerned to quality. A system whose performance levels possess high reliability values ($r_k \approx 1.0$), can derivate in lower repair expenses, lower penalty costs or few guarantee payments. In other systems as medical machines or security systems, a reliable system can save a human life. Thereby, more work about MSS reliability are expected in the next years.

For this dissertation there are some interesting research lines to follow:

- The UGF-Delta method to approximate MSS reliability variances is based on the assumption that the system is composed of independent components. Thereby, the inclusion of dependent measure (e.g. covariance) in the UGF-Delta method is a interesting and important future work.

- In binary systems, there are some proposals to handle the variance as a second objective [14]. For extending this idea to MSS, is necessary to propose a criterion possibly based on Pareto dominance to compare the reliability and variance vectors associated to every MSS.

- The research line mentioned above, involves design reliability problems with two objectives that are solved applying a multi-objective optimization approach. It is a motivation to develop a multi-objective version of the PESO algorithm.

- In this dissertation the RVRP was only applied to the standard VRP. Although, the RVRP can be extended to any VRP variant, there are some specific details associated to every VRP variant that should be studied to efficiently apply the MSS reliability theory. For example, in the capacity VRP variant vehicles with different capacities are used. Thereby, the availability (reliability) of each vehicle can be incorporated to the analysis to design a RVRP that accomplish the customer demands.

- The RVRP avoids unavailable traffic paths using *a priori* information of car accidents, roadworks, traffic congestion, etc. A dynamic RVRP with real time traffic network information should be developed that incorporates the new failures to the model to recalculate the cost-benefit analysis and redesing the route.

Finally, the introduction of MSS reliability theory to other practice problems can improve the modelation of real-world applications. This contributes to narrow the discrepancies between the industry research and academy research (theory and practice problems).

Appendix A

Dirichlet distribution

In probability and statistics, the Dirichlet distribution ($Dir(\alpha)$), is a family of continuous multivariate probability distributions parametrized by a vector α of positive reals. It was introduced by Johann Peter Gustav Lejeune Dirichlet.

The Dirichlet distribution is the multivariate generalization of the beta distribution, and conjugate prior of the categorical distribution and multinomial distribution in Bayesian statistics. That is, its probability density function returns the belief that the probabilities of K rival events are p_i given that each event has been observed $\alpha_i - 1$ times.

A.1 Probability Density Function

The Dirichlet distribution of order $K \geq 2$ with parameters $\alpha_1, \alpha_2, \cdots, \alpha_K > 0$ has a probability density function with respect to Lebesgue measure on the Euclidean space R^{K1} given by

$$f(x_1, x_2, \cdots, x_{K-1}; \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \tag{A.1}$$

$$B(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)}, \qquad \alpha = (\alpha_1, \alpha_2, \cdots, \alpha_K) \tag{A.2}$$

for all $x_1, x_2, \cdots, x_{K-1} > 0$ $x_1 + x_2 + \cdots + x_{K-1} < 1$, where x_K is an abbreviation for $1x_1 - x_2 - \cdots - x_{K-1}$. The density is zero outside this open (K-1)-dimensional simplex.

A.2 Properties

Let $X = (X_1, X_2, \cdots, X_K) \sim Dir(\alpha)$, meaning that the first $K1$ components have the above density and

$$X_K = 1 - X_1 - X_2 - \cdots - X_{K-1} \tag{A.3}$$

Define

$$\alpha_0 = \sum_{i=1}^{K} \alpha_i \tag{A.4}$$

Then

$$E[X_i] = \frac{\alpha_i}{\alpha_0} \tag{A.5}$$

$$Var[X_i] = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)} \tag{A.6}$$

$$Cov[X_i, X_j] = \frac{-\alpha_i\alpha_j}{\alpha_0^2(\alpha_0 + 1)} \tag{A.7}$$

In fact, the marginals are Beta distributions:

$$X_i \sim Beta(\alpha_i, \alpha_0 - \alpha_i) \tag{A.8}$$

A.3 Random Number Generation

A fast method to sample a random vector $x = (x_1, x_2, \cdots, x_K)$ from the K-dimensional Dirichlet distribution with parameters $(\alpha_1, \alpha_2, \cdots, \alpha_K)$ can be resumed as follows:

1. Draw K independent random samples $y_1, y_2, \cdots, y_K$ from gamma distributions, each one with density

$$Gamma(\alpha_i, 1) = \frac{y_i^{\alpha_i - 1} e^{-y_i}}{\Gamma(\alpha_i)} \tag{A.9}$$

2. Define

$$y_0 = \sum_{i=1}^{K} y_i \tag{A.10}$$

3. Finally, set

$$x_i = y_i / y_0 \tag{A.11}$$

.

Appendix B

Multivariate Delta Method

The second order Taylor series expansion of $\hat{r}_k$ at $(\mu_{\hat{p}_{1,1}}, \cdots, \mu_{\hat{p}_{1,n_1}}, \cdots, \mu_{\hat{p}_{N,n_N}})$ is

$$\hat{r}_k(\hat{p}_{1,1}, \cdots, \hat{p}_{N,n_N}) \approx \hat{r}_k(\mu_{\hat{p}_{1,1}}, \cdots, \mu_{\hat{p}_{N,n_N}}) + \sum_{j=1}^{N}\sum_{i=1}^{n_j} \frac{\partial}{\partial \hat{p}_{j,i}} \hat{r}_k(\mu_{\hat{p}_{1,1}}, \cdots, \mu_{\hat{p}_{N,n_N}}) \left[\hat{p}_{j,i} - \mu_{\hat{p}_{j,i}}\right]$$
$$+\frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} 2\frac{\partial^2}{\partial \hat{p}_{j,i}\hat{p}_{h,l}} \hat{r}_k(\mu_{\hat{p}_{1,1}}, \cdots, \mu_{\hat{p}_{N,n_N}}) \left[(\hat{p}_{j,i} - \mu_{\hat{p}_{j,i}})(\hat{p}_{h,l} - \mu_{\hat{p}_{h,l}})\right] \quad \text{(B.1)}$$

For condensing the notation: let F be the reliability function $\hat{r}_k(\hat{p}_{1,1}, \cdots, \hat{p}_{N,n_N})$; let $F_{\hat{p}_{j,i}}$ and $F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}$, the first and second derivatives of $\hat{r}_k(\mu_{\hat{p}_{1,1}}, \cdots, \mu_{\hat{p}_{N,n_N}})$, respectively; and let r_μ be the evaluation of $\hat{r}_k(\mu_{\hat{p}_{1,1}}, \cdots, \mu_{\hat{p}_{N,n_N}})$. Thereby, the next equation condense the second order Taylor series expansion of $\hat{r}_k$:

$$F \approx r_\mu + \sum_{j=1}^{N}\sum_{i=1}^{n_j} F_{\hat{p}_{j,i}} \left[\hat{p}_{j,i} - \mu_{\hat{p}_{j,i}}\right] + \sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}} \left[(\hat{p}_{j,i} - \mu_{\hat{p}_{j,i}})(\hat{p}_{h,l} - \mu_{\hat{p}_{h,l}})\right] \quad \text{(B.2)}$$

The expected value of F is

$$E[F] \approx E\left[r_\mu + \sum_{j=1}^{N}\sum_{i=1}^{n_j} F_{\hat{p}_{j,i}} \left[\hat{p}_{j,i} - \mu_{\hat{p}_{j,i}}\right]\right.$$

$$
\begin{aligned}
&+ \sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})\right]\Bigg] \\
E[F] &= E[r_\mu] + E\left[\sum_{j=1}^{N}\sum_{i=1}^{n_j} F_{\hat{p}_{j,i}}\left[\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}}\right]\right] \\
&+ E\left[\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})\right]\right] \\
E[F] &= r_\mu + \sum_{j=1}^{N}\sum_{i=1}^{n_j} F_{\hat{p}_{j,i}} E\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})\right] \\
&+ \sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}} E\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})\right] E\left[(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})\right] \\
E[F] &= r_\mu + \sum_{j=1}^{N}\sum_{i=1}^{n_j} F_{\hat{p}_{j,i}} \cancelto{0}{E\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})\right]} \\
&+ \sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}} \cancelto{0}{E\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})\right]} \cancelto{0}{E\left[(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})\right]} \\
E[F] &= r_\mu
\end{aligned}
\tag{B.3}
$$

The variance of F is

$$
\begin{aligned}
Var(F) &= E\left[\{F-E[F]\}^2\right] \\
Var(F) &= E\left[\left\{r_\mu + \sum_{j=1}^{N}\sum_{i=1}^{n_j} F_{\hat{p}_{j,i}}\left[\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}}\right]\right.\right. \\
&\left.\left. + \sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})\right] - r_\mu\right\}^2\right] \\
Var(F) &= E\left[\left\{\cancel{r_\mu} + \sum_{j=1}^{N}\sum_{i=1}^{n_j} F_{\hat{p}_{j,i}}\left[\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}}\right]\right.\right. \\
&\left.\left. + \sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})\right] - \cancel{r_\mu}\right\}^2\right]
\end{aligned}
\tag{B.4}
$$

Squaring the sum of products ...

$$
\begin{aligned}
Var(F) &= E\left[\sum_{j=1}^{N}\sum_{i=1}^{n_j} F_{\hat{p}_{j,i}}^2 \left(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}}\right)^2\right. \\
&+ \sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}^2(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})^2(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})^2 \\
&+ 2\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}}F_{\hat{p}_{h,l}}(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}}) \\
&+ 2\sum_{j=1}^{N}\sum_{i=1}^{n_j-1}\sum_{w=i+1}^{n_j} F_{\hat{p}_{j,i}}F_{\hat{p}_{j,w}}(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{j,w}-\mu_{\hat{p}_{j,w}}) \\
&+ 2\sum_{x=1}^{N}\sum_{y=1}^{n_x}\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{x,y}}F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}(\hat{p}_{x,y}-\mu_{\hat{p}_{x,y}})(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{j,w}-\mu_{\hat{p}_{j,w}}) \\
&+ 2\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=1,h\neq j}^{N}\sum_{l=1}^{n_h-1}\sum_{v=l+1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}F_{\hat{p}_{j,i}\,\hat{p}_{h,v}}(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})^2(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})(\hat{p}_{h,v}-\mu_{\hat{p}_{h,v}}) \\
&+ 2\sum_{j=1}^{N}\sum_{i=1}^{n_j-1}\sum_{w=i+1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h-1}\sum_{v=l+1}^{n_h}\left(F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}F_{\hat{p}_{j,w}\,\hat{p}_{h,v}}+F_{\hat{p}_{j,i}\,\hat{p}_{h,v}}F_{\hat{p}_{j,w}\,\hat{p}_{h,l}}\right) \\
&\qquad \left.* (\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{j,w}-\mu_{\hat{p}_{j,w}})(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})(\hat{p}_{h,v}-\mu_{\hat{p}_{h,v}})\right] \qquad \text{(B.5)}
\end{aligned}
$$

Developing the expected value ...

$$
\begin{aligned}
Var(F) &= \sum_{j=1}^{N}\sum_{i=1}^{n_j} F_{\hat{p}_{j,i}}^2 E\left[\left(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}}\right)^2\right] \\
&+ \sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}^2 E\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})^2(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})^2\right] \\
&+ 2\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}}F_{\hat{p}_{h,l}} E\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})\right] \\
&+ 2\sum_{j=1}^{N}\sum_{i=1}^{n_j-1}\sum_{w=i+1}^{n_j} F_{\hat{p}_{j,i}}F_{\hat{p}_{j,w}} E\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{j,w}-\mu_{\hat{p}_{j,w}})\right]
\end{aligned}
$$

$$+2\sum_{x=1}^{N}\sum_{y=1}^{n_x}\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{x,y}}F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}E\left[(\hat{p}_{x,y}-\mu_{\hat{p}_{x,y}})(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{j,w}-\mu_{\hat{p}_{j,w}})\right]$$
$$+2\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=1,h\neq j}^{N}\sum_{l=1}^{n_h-1}\sum_{v=l+1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}F_{\hat{p}_{j,i}\,\hat{p}_{h,v}}E\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})^2(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})(\hat{p}_{h,v}-\mu_{\hat{p}_{h,v}})\right]$$
$$+2\sum_{j=1}^{N}\sum_{i=1}^{n_j-1}\sum_{w=i+1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h-1}\sum_{v=l+1}^{n_h}\left(F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}F_{\hat{p}_{j,w}\,\hat{p}_{h,v}}+F_{\hat{p}_{j,i}\,\hat{p}_{h,v}}F_{\hat{p}_{j,w}\,\hat{p}_{h,l}}\right)$$
$$*E\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{j,w}-\mu_{\hat{p}_{j,w}})(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})(\hat{p}_{h,v}-\mu_{\hat{p}_{h,v}})\right] \tag{B.6}$$

Calculating the expected value for every term ...

$$Var(F)=\sum_{j=1}^{N}\sum_{i=1}^{n_j}F_{\hat{p}_{j,i}}^2Var(\hat{p}_{j,i})$$
$$+\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h}F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}^2Var(\hat{p}_{j,i})Var(\hat{p}_{h,l})$$
$$+2\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h}F_{\hat{p}_{j,i}}F_{\hat{p}_{h,l}}E\left[(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{h,l}-\mu_{\hat{p}_{h,l}})\right]\to 0$$
$$+2\sum_{j=1}^{N}\sum_{i=1}^{n_j-1}\sum_{w=i+1}^{n_j}F_{\hat{p}_{j,i}}F_{\hat{p}_{j,w}}Cov(\hat{p}_{j,i},\hat{p}_{j,w})$$
$$+2\sum_{x=1}^{N}\sum_{y=1}^{n_x}\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h}F_{\hat{p}_{x,y}}F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}E\left[(\hat{p}_{x,y}-\mu_{\hat{p}_{x,y}})(\hat{p}_{j,i}-\mu_{\hat{p}_{j,i}})(\hat{p}_{j,w}-\mu_{\hat{p}_{j,w}})\right]\to 0$$
$$+2\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=1,h\neq j}^{N}\sum_{l=1}^{n_h-1}\sum_{v=l+1}^{n_h}F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}F_{\hat{p}_{j,i}\,\hat{p}_{h,v}}Var(\hat{p}_{j,i})Cov(\hat{p}_{h,l},\hat{p}_{h,v})$$
$$+2\sum_{j=1}^{N}\sum_{i=1}^{n_j-1}\sum_{w=i+1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h-1}\sum_{v=l+1}^{n_h}\left(F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}F_{\hat{p}_{j,w}\,\hat{p}_{h,v}}+F_{\hat{p}_{j,i}\,\hat{p}_{h,v}}F_{\hat{p}_{j,w}\,\hat{p}_{h,l}}\right)$$
$$*Cov(\hat{p}_{j,i},\hat{p}_{j,w})Cov(\hat{p}_{h,l},\hat{p}_{h,v}) \tag{B.7}$$

Finally ...

$$
\begin{aligned}
Var(F) &= \sum_{j=1}^{N}\sum_{i=1}^{n_j} F_{\hat{p}_{j,i}}^2 Var(\hat{p}_{j,i}) \\
&+ \sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}}^2 Var(\hat{p}_{j,i})Var(\hat{p}_{h,l}) \\
&+ 2\sum_{j=1}^{N}\sum_{i=1}^{n_j-1}\sum_{w=i+1}^{n_j} F_{\hat{p}_{j,i}} F_{\hat{p}_{j,w}} Cov(\hat{p}_{j,i},\hat{p}_{j,w}) \\
&+ 2\sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=1,h\neq j}^{N}\sum_{l=1}^{n_h-1}\sum_{v=l+1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}} F_{\hat{p}_{j,i}\,\hat{p}_{h,v}} Var(\hat{p}_{j,i})Cov(\hat{p}_{h,l},\hat{p}_{h,v}) \\
&+ 2\sum_{j=1}^{N}\sum_{i=1}^{n_j-1}\sum_{w=i+1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h-1}\sum_{v=l+1}^{n_h} \left(F_{\hat{p}_{j,i}\,\hat{p}_{h,l}} F_{\hat{p}_{j,w}\,\hat{p}_{h,v}} + F_{\hat{p}_{j,i}\,\hat{p}_{h,v}} F_{\hat{p}_{j,w}\,\hat{p}_{h,l}}\right) \\
&\quad * Cov(\hat{p}_{j,i},\hat{p}_{j,w})Cov(\hat{p}_{h,l},\hat{p}_{h,v})
\end{aligned}
\tag{B.8}
$$

The covariance of two system reliabilities, $\hat{r}_a$ and $\hat{r}_b$, represented by the two variables, F and G, respectively; is approximated by the following expression:

$$
\begin{aligned}
Cov(F,G) &= \sum_{j=1}^{N}\sum_{i=1}^{n_j} F_{\hat{p}_{j,i}} G_{\hat{p}_{j,i}} Var(\hat{p}_{j,i}) \\
&+ \sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h} F_{\hat{p}_{j,i}\,\hat{p}_{h,l}} G_{\hat{p}_{j,i}\,\hat{p}_{h,l}} Var(\hat{p}_{j,i})Var(\hat{p}_{h,l}) \\
&+ \sum_{j=1}^{N}\sum_{i=1}^{n_j-1}\sum_{w=i+1}^{n_j} \left[F_{\hat{p}_{j,i}} G_{\hat{p}_{j,w}} + G_{\hat{p}_{j,i}} F_{\hat{p}_{j,w}}\right] Cov(\hat{p}_{j,i},\hat{p}_{j,w}) \\
&+ \sum_{j=1}^{N}\sum_{i=1}^{n_j}\sum_{h=1,h\neq j}^{N}\sum_{l=1}^{n_h-1}\sum_{v=l+1}^{n_h} \left[F_{\hat{p}_{j,i}\,\hat{p}_{h,l}} G_{\hat{p}_{j,i}\,\hat{p}_{h,v}} + G_{\hat{p}_{j,i}\,\hat{p}_{h,l}} F_{\hat{p}_{j,i}\,\hat{p}_{h,v}}\right] \\
&\quad * Var(\hat{p}_{j,i})Cov(\hat{p}_{h,l},\hat{p}_{h,v}) \\
&+ \sum_{j=1}^{N}\sum_{i=1}^{n_j-1}\sum_{w=i+1}^{n_j}\sum_{h=j+1}^{N}\sum_{l=1}^{n_h-1}\sum_{v=l+1}^{n_h} \left[\left(F_{\hat{p}_{j,i}\,\hat{p}_{h,l}} G_{\hat{p}_{j,w}\,\hat{p}_{h,v}} + F_{\hat{p}_{j,i}\,\hat{p}_{h,v}} G_{\hat{p}_{j,w}\,\hat{p}_{h,l}}\right)\right. \\
&\quad \left. + \left(G_{\hat{p}_{j,i}\,\hat{p}_{h,l}} F_{\hat{p}_{j,w}\,\hat{p}_{h,v}} + G_{\hat{p}_{j,i}\,\hat{p}_{h,v}} F_{\hat{p}_{j,w}\,\hat{p}_{h,l}}\right)\right] * Cov(\hat{p}_{j,i},\hat{p}_{j,w})Cov(\hat{p}_{h,l},\hat{p}_{h,v})
\end{aligned}
\tag{B.9}
$$

.

.

Bibliography

[1] Ando N, Taniguchi E. Travel time reliability in vehicle routing and scheduling with time windows. *Networks and Spatial Economics*, 6(3-4):293–311, 2006.

[2] Angeline P. Evolutionary optimization versus particle swarm optimization: philosophy and performance differences. In: *Proceedings of the Seventh Annual Conference on Evolutionary Programming, EP VII*, 601–610, San Diego, 1998.

[3] Aven T. On performance measures for multistate monotone systems. *Reliability Engineering and System Safety*, 41:259–266, 1993.

[4] Blackwell T. Particle swarms and population diversity. *Soft Computing*, 9(11):793-802, 2005.

[5] Barlow R, Wu A. Coherent systems with multi-state components. *Mathematics of Operations Research*, 3:275–281, 1978.

[6] BS 4778 British Standard. Glossary of terms used in quality assurance including reliability and maintainability terms. British Standard Institution, London.

[7] Cagnina L, Esquivel S, Coello C. A particle swarm optimizer for constrained numerical optimization. In: *Proceedings of the 9th International Conference - Parallel problem Solving from Nature, PPSN IX*, 910-919, Jeykjavik, 2006.

[8] Caldarola L. Coherent systems with multi-state elements. *Nuclear Engineering and Design*, 58:127–139, 1980.

[9] Chern M. On the computational complexity of reliability redundancy allocation in a series system. *Operations Research Letters*, 11:309–315, 1992.

[10] Coath G, Halgamuge S. A comparison of constraint-handling methods for the application of particle swarm optimization to constrained nonlinear optimization problems. In:

Proceedings of the 2003 Congress on Evolutionary Computation, 2419–2425, Canberra, 2003.

[11] Coello C. Theoretical and numerical constraint handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, 2002.

[12] Coit D, Smith A. Reliability optimization of series-parallel systems using a genetic algorithm. *IEEE Transactions on Reliability*, 45(2):254–260, 1996.

[13] Coit D. System reliability confidence intervals for complex systems with estimated component reliability. *IEEE Transactions on Reliability*, 46(4):487–493, 1997.

[14] Coit D, Jin T. Multi-criteria optimization: maximization of a system reliability estimate and minimization of the estimate variance. In: *Proceedings of the 2001 European Safety and Reliability International Conference, ESREL*, Turin, 2001.

[15] Cooch E, White G. A Gentle Introduction to Program MARK. Ithaca, New York, 2009.

[16] Das S, Konar A, Chakraborty U. Improving particle swarm optimization with differentially perturbed velocity. In: *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, 177–184, Washington, 2005.

[17] Dantzig G, Ramser R. The truck dispatching problem. *Management Science*, 6:80–91, 1959.

[18] Deb K. An efficient constraint handling method for genetic algorithms. *Computer Methods in Appplied Mechanics and Engineering*, 186(2-4):311–338, 2000.

[19] Ebeling C. An introduction to reliability and maintainability engineering. Boston: McGraw-Hill, 1997.

[20] Elbeltagi E, Hegazy T, Grierson D. Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics*, 19:43–53, 2005.

[21] Eberhart R, Dobbins R, Simpson P. Computational intelligence PC tools. San Diego: Academic Press Professional, 1996.

[22] Eberhart R, Shi Y. Comparison between genetic algorithms and particle swarm optimization. In: *Proceedings of the Seventh Annual Conference on Evolutionary Programming, EP VII*, 611–616, San Diego, 1998.

[23] Engelbrecht A. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, 2005.

[24] Ester M, Kriegel H, Sander J. Algorithms and applications for spatial data mining. In: Miller H, Han J, Editors, *Geographic Data Mining and Knowledge Discovery*, 160–187, Taylor & Francis, 2001.

[25] Fan Y, Kalaba R, Moore J. Arriving on time. *Journal of Optimization Theory and Applications*, 127:497-513, 2005.

[26] Fisher R. The correlation between relatives on the supposition of mendelian inheritance. *Philosophical Transactions of the Royal Society of Edinburgh*, 52:399-433, 1918.

[27] Fogel D. An introduction to simulated evolutionary optimization. *IEEE Transaction on Neural Networks*, 5(1):3–14, 1994.

[28] García-Villoria A, Pastor R. Introducing dynamic diversity into a discrete particle swarm optimization. *Computers and Operations Research*, 36(3):951–956, 2009.

[29] Golden B., Assad A, Wasil E. Routing vehicles in the real world: applications in the solid waste, beverage, food, dairy, and newspaper industries. In: Toth P, Vigo D, Editors, *The Vehicle Routing Problem*, 245–286, SIAM, 2002.

[30] Goodman L. On the exact variance of products. *Journal of the American Statistical Association*, 50(292):708–713, 1960.

[31] Greene W. Econometric Analysis. New Jersey: Prentice Hall, 2002.

[32] Griffith W. Multistate reliability models. *Journal of Applied Probability*, 17:735-744, 1980.

[33] Gurov S, Utkin V. Cold standby systems with imperfect and non-instantaneous switch-over mechanism. *Microelectronics and Reliability*, 36(10):1425–1438, 1996.

[34] Hamida S, Petrowski A. The need for improving the exploration operators for constrained optimization problems. In: *Proceedings of the Congress on Evolutionary Computation*, 1176–1183, San Diego, 2000.

[35] He S, Prempain E, Wu Q. An improved particle swarm optimizer for mechanical design optimization problems. *Engineering Optimization*, 36(5):585–605, 2004.

[36] Hernandez-Aguirre A, Botello S, Coello C. PASSSS: an implementation of a novel diversity strategy to handle constraints. In: *Proceedings of the 2004 Congress on Evolutionary Computation, CEC 2004*, 403–410, Portland, 2004.

[37] Hernandez-Aguirre A, Botello S, Coello C, Lizarraga G, Mezura E. Handling constraints using multiobjective optimization concepts. *International Journal for Numerical Methods in Engineering*, 59(13):1989–2017, 2004.

[38] Hernández-Aguirre A, Muñoz-Zavala A, Villa-Diharce E, Botello S. COPSO: constrained optimization via PSO algorithm. *Technical Report of the Computer Sciences Department*, Centro de Investigación en Matemáticas, Guanajuato, México, 2007. *http://www.cimat.mx/reportes/enlinea/I-07-04.pdf*

[39] Holland J. Adaptation in natural and artificial systems. Ann Arbor, MI: The University of Michigan Press, 1975.

[40] Hoyland A, Rausand M. System Reliability Theory: Models and Statistics Methods. New York: John Wiley & Sons, 1994.

[41] Hu X, Eberhart R. Solving constrained nonlinear optimization problems with particle swarm optimization. In: *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics, SCI 2002*, 203–206, Orlando, 2002.

[42] Hu X, Eberhart R, Shi Y. Engineering optimization with particle swarm. In: *Proceedings of the IEEE Swarm Intelligence Symposium*, 53–57, Indianapolis, 2003.

[43] Huang J, Zuo M, Wu Y. Generalized multi-state k-out-of-n:G systems. *IEEE Transactions on Reliability*, 49(1):105–111, 2000.

[44] Hudson J, Kapur K. Reliability theory for multistate systems with multistate elements. *Microelectronics Reliability* 22:1–7, 1982.

[45] Hwang F, Yao Y. Multistate consecutively-connected systems. *IEEE Transactions on Reliability*, 38(4):472–474, 1989.

[46] Innocente M, Sienz J. Constraint-handling techniques for particle swarm optimization algorithms. In: *Proceedings of the 7th ASMO UK/ISSMO Conference on Engineering Design Optimization,*, 203–223, Bath, 2008.

[47] ISO 8402 International Standard. Quality vocabulary. International Standards Organization.

[48] Jie J, Zeng J. Particle swarm optimization with diversity-controlled acceleration coefficients. In: *Proceedings of the Third International Conference on Natural Computation, 2007, ICNC 2007*, 150–154, Haikou, 2007.

[49] Jin T. Complex system reliability estimation and optimization considering component estimation uncertainties. Ph.D. Thesis, Rutgers University, Piscataway, USA, 2001.

[50] Jin T, Coit D. Variance of system reliability estimation with arbitrarily repeated components. *IEEE Transactions on Reliability*, 50(4):409–413, 2001.

[51] Jin T, Coit D. Unbiased variance estimates for system reliability estimate using block decompositions. *IEEE Transactions on Reliability*, 57(3):458–464, 2008.

[52] Jin T. Hierarchical variance decomposition of system reliability with duplicated components. *IEEE Transactions on Reliability*, 57(4):564–573, 2008.

[53] Jing-An L, Yue W, Kin K, Ke L. Reliability estimation and prediction of multi-state components and coherent systems. *Reliability Engineering and System Safety*, 88:93-98, 2005.

[54] Johnson N, Kotz S, Kemp A. Univariate Discrete Distributions. New York, John Wiley and Sons, 1992.

[55] Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of the IEEE International Conference On Neural Networks*, 4:1942–1948, 1995.

[56] Kennedy J. The particle swarm: social adaptation of knowledge. In: *Proceedings of the IEEE International Conference on Evolutionary Computation*, 303–308, 1997.

[57] Kennedy J, Eberhart R. The particle swarm: social adaptation in information-processing systems. London: McGraw-Hill, 1999.

[58] Kennedy J, Mendes R. Population structure and particle swarm performance. In: *Proceedings of the 2002 Congress on Evolutionary Computation*, 1671–1676, Honolulu, 2002.

[59] Krink T, Vesterstrom J, Riget J. Particle swarm optimization with spatial particle extension. In: *Proceedings of the 2002 Congress on Evolutionary Computation*, 1474–1479, Honolulu, 2002.

[60] Kuo W, Zhang W, Zuo M. A consecutive-k-out-of-n:G system: the mirror image of a consecutive-k-out-of-n:F system. *IEEE Transactions on Reliability*, 39(2):244–253, 1990.

[61] Kuo W, Prasad R. An annotated overview of system reliability optimization. *IEEE Transactions on Reliability*, 49(2):176–187, 2000.

[62] Kuo T, Chen S, Chiu J. Tabu search in the redundancy allocation optimization for multi-state series-parallel systems. *Journal of the Chinese Institute of Industrial Engineers*, 24:210–218, 2007.

[63] Lenstra J, Rinnooy A. Complexity of vehicle routing and scheduling problems. *Networks*, 11:221-227, 1981.

[64] Levitin G, Lisnianski A, Ben-Haim H, Elmakis D. Redundancy optimization for series-parallel multi-state systems. *IEEE Transactions on Reliability*, 47(2):165–172, 1998.

[65] Levitin G, Lisnianski A. Joint redundancy and maintenance optimization for multi-state series-parallel systems. *Reliability Engineering and System Safety*, 64:33–42, 1999.

[66] Levitin G, Lisnianski A. Optimization of imperfect preventive maintenance for multi-state systems. *Reliability Engineering and System Safety*, 67:193–203, 2000.

[67] Levitin G, Lisnianski A, Ben-Haim H, Elmakis D. Genetic algorithm and universal generating function technique for solving problems of power system reliability optimization. In: *Proceedings of the International Conference on Electric Utility Deregulation and Restructuring and Power Technologies 2000, DRPT 2000*, 582–585, London, 2000.

[68] Levitin G, Lisnianski A. Optimal replacement scheduling in multi-state series-parallel systems. *Quality and Reliability Engineering International*, 16:157–162, 2000.

[69] Levitin G, Lisnianski A. Survivability maximization for vulnerable multi-state systems with bridge topology. *Reliability Engineering and System Safety*, 70:125–140, 2000.

[70] Levitin G, Lisnianski A. Reliability optimization for weighted voting system. *Reliability Engineering and System Safety*, 71:131-138, 2001.

[71] Levitin G, Lisnianski A. Structure optimization of multi-state system with two failure modes. *Reliability Engineering and System Safety*, 72:75-89, 2001.

[72] Levitin G, Lisnianski A. A new approach to solving problems of multi-state system reliability optimization. *Quality and Reliability Engineering International*, 17:93-104, 2001.

[73] Levitin G. Reliability evaluation for acyclic consecutively connected networks with multistate elements. *Reliability Engineering and System Safety*, 73(2):137–143, 2001.

[74] Levitin G, Lisnianski A. Multi-state system reliability analysis and optimization. In: Pham H, editor, *Handbook of Reliability Engineering*, 61–90, Springer-Verlag, 2008.

[75] Levitin G. A universal generating function approach for the analysis of multi-state systems with dependent elements. *Reliability Engineering and System Safety*, 84(3):285–292, 2004.

[76] Levitin G. Universal Generating Function in Reliability Analysis and Optimization. Springer Series in Reliability Engineering, London, Springer-Verlag, 2005

[77] Levitin G, Hu X, Dai Y. Particle swarm optimization in reliability engineering. In: Levitin G, editor, *Computational Intelligence in Reliability Engineering. New Metaheuristics, Neural and Fuzzy Techniques in Reliability*, Series: Studies in Computational Intelligence, 40:83–112, Springer-Verlag, 2006.

[78] Levitin G. Universal generating function in analysis of multi-state systems. In: Misra K, editor, *Handbook of Performability Engineering*, 447–464, Springer-Verlag, 2008.

[79] Liang J, Suganthan P. Dynamic multi-swarm particle swarm optimizer with a novel constraint-handling mechanism. In: *Proceedings of the 2006 IEEE Congress on Evolutionary Computation, CEC2006*, 316–323, Vancouver, 2006.

[80] Liang J, Runarsson T, Mezura-Montes E, Clerc M, Suganthan P, Coello C, Deb K. Problem definitions and evaluation criteria for the CEC 2006. In: *Special Session on Constrained Real-Parameter Optimization*, Technical Report, 2006.

[81] Lisnianski A, Levitin G, Ben-Haim H. Structure optimization of multi-state system with time redundancy. *Reliability Engineering and System Safety*, 67:103–112, 2000.

[82] Lisnianski A, Levitin G. Universal generating function application to multi-state system reliability analysis & optimization. In: *Abstracts' book of the Second International Conference on Mathematical Methods in Reliability, MMR2000*, 715–718, Bordeaux, 2000.

[83] Lisnianski A, Levitin G. Multi-state system reliability: assessment, optimization and applications. Series on Quality, Reliability and Engineering Statistics, London, World Scientific, 2003.

[84] Liu X, Su J, Han Y. An improved particle swarm optimization for traveling salesman problem. In: *Proceedings of the Third International Conference on Intelligent Computing, ICIC 2007*, 803–812, Qingdao, 2007.

[85] Lovbjerg M, Rasmussen T, Krink T. Hybrid particle swarm optimiser with breeding and subpopulations. In: *Proceedings of the 2001 Genetic and Evolutionary Computation Conference*, 469–476, San Francisco, 2001.

[86] Malinowski J, Preuss W. Reliability evaluation for tree-structured systems with multistate components. *Microelectronics Reliability*, 36(1):9–17, 1996.

[87] Malinowski J, Preuss W. Reliability of reverse-tree-structured systems with multistate components. *Microelectronics Reliability*, 36(1):1–7, 1996.

[88] Marseguerra M, Zio E, Podofillini L, Coit D. Optimal design of reliable network systems in presence of uncertainty. *IEEE Transactions on Reliability*, 54(2):243–253, 2005.

[89] Mezura E. Alternatives to handle constraints in evolutionary optimization. Ph.D. Thesis, CINVESTAV-IPN, D.F., Mexico, 2004.

[90] Mezura E, Coello C. A survey of constraint-handling techniques based on evolutionary multiobjective optimization. Technical Report EVOCINV-04-2006, Evolutionary Computation Group at CINVESTAV, Departamento de Computacin, CINVESTAV-IPN, Mxico, October 2006.

[91] Michalewicz Z. A Survey of Constraint Handling Techniques in Evolutionary Computation Methods. In: *Proceedings of the 4th Annual Conference on Evolutionary Programming*, 135–155, San Diego, 1995.

[92] Muñoz-Zavala A, Villa-Diharce E, Hernández-Aguirre A. Particle evolutionary swarm for design reliability optimization. In: *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005*, 856–869, Guanajuato, 2005.

[93] Muñoz-Zavala A, Hernández-Aguirre A, Villa-Diharce E. Constrained optimization via particle evolutionary swarm optimization algorithm (PESO). In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2005*, 209–216, Washington, 2005.

[94] Muñoz-Zavala A, Hernández-Aguirre A, Villa-Diharce E. Robust PSO-based constrained optimization by perturbing the particle's memory. In: Chan F, Tiwari M, Editors, *Swarm*

Intelligence, Focus on Ant and Particle Swarm Optimization, 57–76, Vienna: I-Tech Education and Publishing, 2007.

[95] Muñoz-Zavala A, Hernández-Aguirre A, Villa-Diharce E, Botello-Rionda S. Constrained optimization with an improved particle swarm optimization algorithm. *International Journal of Intelligent Computing and Cybernetics*, 1(3):425–453, 2008.

[96] Muñoz-Zavala A, Hernández-Aguirre A, Villa-Diharce E. Particle evolutionary swarm for continuous optimization problems (PESCOP). In: *Proceedings of Cuarto Congreso Mexicano de Computación Evolutiva, COMCEV 2008*, 27–32, Guanajuato, 2008.

[97] Muñoz-Zavala A, Hernández-Aguirre A, Villa-Diharce E. Continuous constrained optimization with dynamic tolerance using the COPSO algorithm. In: Mezura-Montes E, Editor, *Constraint-handling in evolutionary optimization*, 1–23, Springer, ISBN: 978-3-642-00618-0, 2009.

[98] Murchland J. Fundamental concepts and relations for reliability analysis of multistate systems, reliability and fault tree analysis. *Theoretical and Applied Aspects of System Reliability*, 581–618, SIAM, 1975.

[99] Natvig B. Two suggestions of how to define a multistate coherent system. *Advances in Applied Probability*, 14:434–455, 1982.

[100] Natvig B. Multistate coherent systems. *Encyclopedia of Statistical Sciences*, 5:732-735, 1985.

[101] Natvig B, Sormo S, Holen A, Hogasen G. Multistate reliability theory a case study. *Advances in Applied Probability*, 18:921–932, 1986.

[102] Natvig B, Morch H. An application of multistate reliability theory to an offshore gas pipeline network. *International Journal of Reliability, Quality and Safety Engineering*, 10:361–381, 2003.

[103] Natvig B. Multi-state reliability theory. *Statistical Research Report No. 1*, Department of Mathematics, University of Oslo, Oslo, Norway, 2007.

[104] Nourelfath M, Nahas N. Ant colony optimization to redundancy allocation for multi-state systems. In: *Proceedings of the Fourth International Conference on Mathematical Methods in Reliability Methodology and Practice*, Santa Fe, 2004.

[105] Ochoa A, Muñoz-Zavala A, Hernández-Aguirre A. A hybrid system using PSO and data mining for determining the ranking of a new participant in Eurovision. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2008*, 1713–1714, Atlanta, 2008.

[106] Oehlert G. A Note on the Delta Method. *The American Statistician*, 46(1):27–29, 1992.

[107] Osman I. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problems. *Annals of Operations Research*, 41:421–452, 1993.

[108] Parsopoulos K, Vrahatis M. Particle swarm optimization method for constrained optimization problems. *Intelligent Technologies - Theory and Application: New Trends in Intelligent Technologies*, 76:214–220, 2002.

[109] Parsopoulos K, Vrahatis M. Unified particle swarm optimization for solving constrained engineering optimization problems. In: *Proceedings of the International Conference on Advances in Natural Computation*, 582–591, Changsha, 2005.

[110] Pisinger D, Ropke S. A general heuristic for vehicle routing problems. *Computers and Operations Research*, 34(8):2403–2435, 2007.

[111] Price K, Storn R, Lampinen J. Differential evolution: a practical approach to global optimization. Berlin: Springer Verlag, 2005.

[112] Ramirez-Marquez J, Coit D. A heuristic for solving the redundancy allocation problem for multi-state series-parallel systems. *Reliability Engineering and System Safety*, 83(3):341–349, 2004.

[113] Ramirez-Marquez J, Jiang W. On improved confidence bounds for system reliability. *IEEE Transactions on Reliability*, 55(1):26–36, 2006.

[114] Ramirez-Marquez J. New approaches for reliability design in multistate systems. In: Misra K, editor, *Handbook of Performability Engineering*, 465–476, Springer-Verlag, 2008.

[115] Ramirez-Marquez J, Levitin G. Algorithm for estimating reliability confidence bounds of multi-state system. *Reliability Engineering and System Safety*, 93(8):1231–1243, 2008.

[116] Riget J, Vesterstrom J. A diversity-guided particle swarm optimizer - the ARPSO. Technical Report 2002-02, Department of Computer Science, University of Aarhus, 2002.

[117] Rizzoli A, Oliverio F, Montemanni R, Gambardella L. Ant colony optimisation for vehicle routing problems: from theory to applications. Technical Report IDSIA-15-04, *Istituto Dalle Molle di Studi sull'Intelligenza Artificiale*, 2004.

[118] Rocco C, Zio E. Solving advanced network reliability problems by means of cellular automata and Monte Carlo sampling. *Reliability Engineering and System Safety*, 89(2):219–226, 2005.

[119] Rosenbaltt J. Confidence Limits for Reliability of Complex Systems. In: Zelen M, editor, *Statistical Theory of Reliability*, University of Wisconsin Press, 1963.

[120] Settles M, Soule T. Breeding swarms: a GA/PSO hybrid. In: *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, 161–168, Washington, 2005.

[121] Soh S. An efficient cutset approach for evaluating communication network reliability with heterogeneous link-capacities. *IEEE Transactions on Reliability*, 54(1):133–144, 2005.

[122] Storn R, Price K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical Report TR-95-012*, International Computer Science Institute, 1995.

[123] Storn R. System design by constraint adaptation and differential evolution. *IEEE Transactions on Evolutionary Computation*, 3(1):22–34, 1999.

[124] Taboada H, Espiritu J, Coit D. Design allocation of multi-state series-parallel systems for power systems planning: a multiple objective evolutionary approach. *Journal of Risk & Reliability*, 222(3):381–391, 2008.

[125] Taboada H, Espiritu J, Coit D. MOMS-GA: a multi-objective multi-State genetic algorithm for system reliability optimization design problems. *IEEE Transactions on Reliability*, 57(1):182–191, 2008.

[126] Thangiah S. Vehicle routing with time windows using genetic algorithms. In: Chambers L, Editor, *Application Handbook of Genetic Algorithms: New Frontiers Vol. II*, 253-277, Boca Raton:CRC Press, 1995.

[127] Tian Z, Zuo M, Huang H. Reliability-redundancy allocation for multi-state series-parallel systems. *IEEE Transactions on Reliability*, 57(2):303–310, 2008.

[128] Toscano G, Coello C. A constraint-handling mechanism for particle swarm optimization. In: *Proceedings of the 2004 Congress on Evolutionary Computation*, 1396–1403, Portland, 2004.

[129] Toth P, Vigo D. The vehicle routing problem. Philadelphia: SIAM Monographs on Discrete Mathematics and Applications, 2002.

[130] Ushakov I. A universal generating function. *Soviet Journal of Computer and Systems Sciences*, 24:37-49, 1986.

[131] Ushakov I. Reliability analysis of multi-state systems by means of a modified generating function. *Journal of Information Processing and Cybernetics*, 34:24–29, 1988.

[132] Van den Bergh F. An analysis of particle swarm optimizers. Ph.D. Thesis, University of Pretoria, South Africa, 2002.

[133] Van Woensel T, Kerbache L, Peremans H, Vandaele N. A queueing framework for routing problems with time-dependent travel times. *Journal of Mathematical Modelling and Algorithms*, 6(1):151–173, 2007.

[134] Van Woensel T, Kerbache L, Peremans H, Vandaele N. Vehicle routing with dynamic travel times: a queueing approach. *European Journal of Operational Research 2008*, 186(3):990–1007, 2008.

[135] Wang K, Huang L, Zhou C, Pang W. Particle swarm optimization for traveling salesman problem. In: *Proceedings of the International Conference on Machine Learning and Cybernetics 2003*, 1583-1585, Xi-an, 2003.

[136] Wilf H. Generatingfunctionology. San Diego, Academic Press, 1994.

[137] Wolpert D, Macready W. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[138] Wood A. Multistate block diagrams and fault trees. *IEEE Transactions on Reliability*, 34:236–240, 1985.

[139] Xu J, Long S. Using the Delta Method to Construct Confidence Intervals for Predicted Probabilities, Rates, and Discrete Changes. Technical Report, *Indiana University*, Bloomington, USA, 2005.

[140] Yeh, W. A simple algorithm to search for all MCs in networks. *European Journal of Operational Research*, 174(3):1694–1705, 2006.

[141] Yeh, W. A simple universal generating function method for estimating the reliability of general multi-state node networks. *IIE Transactions*, 41(1):3–11, 2009.

[142] Zang X, Sun H, Trivedi K. A BDD-based algorithm for reliability analysis of phased-mission systems. *IEEE Transactions on Reliability*, 48(1):50–60, 1999.

[143] Zhang J, Xie F. DEPSO: hybrid particle swarm with differential evolution operator. In: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 3816–3821, Washington, 2003.

[144] http://personales.upv.es/arodrigu/rutas/documentos.htm

[145] http://earth.google.com/