# *Contributions to Statistical Machine Learning Algorithm*



YanHong Cui

PhD Thesis Submitted on 05 December 2011

PhD Thesis Presented for the Degree of Doctor of Philosophy, in the Department of Statistical Sciences, University of Cape Town, South Africa.

Supervisor: Professor Renkuan Guo

# Chapter 1. Introduction

## 1.1 Statistical Machine Learning Algorithm Development

This thesis's research focus is on computational statistics along with DEAR (abbreviation of differential equation associated regression) model direction, and that in mind, the journal papers are written as contributions to statistical machine learning algorithm literature.

In the previous research experiences (MSc thesis “Statistical-Grey Consistent Grey Differential Equation Modelling”), one is made aware that any statistical analysis and modelling including modelling parameter estimation, hypothesis inferences and confidence interval constructions, is an optimization with respect to the given or pre-specified criterion, subject to some constraints over the collected data. In other words, mathematically speaking, statistical modelling is to search the optimal solution of an objective function subject to a set of constraints given the data information. However, such an optimization process, without efficient computational methodology to materialize it in numerical form, is merely a bunch of theoretical results, an empty promise in some sense.

A new field in modern statistics, the computational statistics, including data mining, has already opened, in which model selections and efficient computations are organically combined. It is obvious that the research consciousness has not been narrowed to the degree of the statistical machine learning algorithm yet.

The thesis research process evolved gradually but the process can be divided into three developing stages.

1. The first one is λ-algorithm creation and application stage. Computations on grey differential equation models and later on DEAR models were carried on from the MSc thesis. One is conscious that the serious problems of the evolutionary algorithms, particularly, GA (abbreviation of genetic algorithm), in computations of fitting grey differential equation models and therefore the improvements of GA (in Matlab) is the first task. But it is not possible to improve it, because in GA searching scheme the string element set $\Theta = \{0,1\}$ only has two elements, and thus the searching process is not be able to carry more information for classifications or clustering, so that the uncertain behaviour is difficult to analyze. Then increasing the number of the elements in element set $\Theta$ appeared in my mind. This led to the

development of the first paper (Cui et al., 2009). The partial success of five-element string algorithm led to further efforts in improving the new algorithm. In order for the new algorithm to be accepted by scientific communities, the new algorithm is named as the λ-global optimization algorithm (λ-algorithm in short or λ-scheme later). The λ-algorithm is combined with Nash-equilibrium for searching bi-level programs (named as Nash-λ algorithm), and also maximum likelihood procedure (named as maximum likelihood λ-algorithm). During this stage, two issues came to focus: (1) the uncertain behaviour has to be explored under certain measure theoretic framework; (2) model selections are of uncertain statistical decision nature.

2. The second one is addressing the above identified two issues. The stochastic behaviour failed to explain the mechanism that governs the SML, and so the research focus is turned to Liu's uncertainty measure theory (2007, 2010, 2011) for creating uncertain statistical decision theory, which has even created a version of uncertain Bayes formulae. The computations is largely touching the finding of supremum or infimum of set functions, which provides rich computation experiences and benefit the later uncertain canonical process regressions. This led to two journal papers on uncertain decision theory.
3. The third stage is to address the DEAR modelling by a single measure foundation. Guo et al. (2011) first created uncertain canonical process regression models, including uncertain canonical process DEAR models, then, built up Bayesian DEAR models. The journal paper "Probabilistic DEAR models" is written under the invitation of the Chief Editor, Professor Wang of the International Journal of Machine Learning and Cybermetric, in which the Gaussian process DEAR models is merged with λ-global optimization scheme into a new statistical machine learning algorithm.



## 1.2 Aims and Objectives

The overall objective of this thesis is:

- To use various computation methods, to combine Gaussian process DEAR model selections with λ-global optimization scheme, into a new Statistical Machine Learning algorithm (DEAR λ-algorithm), in order to deal with data mining problems.

The aims of the thesis can be classified into 3 broad categories:

- To use statistical and mathematical methodology to design a new optimization scheme (global optimization algorithm), λ-algorithm.
- To build up uncertainty decision theory and modelling to deal with decision problem with general uncertainty data.
- To build up probabilistic DEAR modelling families and merges with λ-global optimization scheme such that a new statistical machine learning algorithm, DEAR λ-algorithm is created.

Lambda algorithm aims:

- To develop and explore a new global optimization algorithm-lambda algorithm
- To develop and explore the mathematical foundation and statistical explanation for the lambda algorithm.
- To develop and explore Nash-lambda algorithm to deal with bi-level non-cooperative game models optimization problems.
- To apply the lambda algorithm into maximum likelihood estimation methods.
- To develop the dual model mutual operation to accelerate the convergence speed of the optimization.
- To develop the 3-states markov decision models for different information automatic classification
- To develop shrinking searching domain operation to allow the lambda algorithm only use 3 or 4 bits to represent each variable.
- To develop the lambda algorithm allowed optimize the objective function under various constrains.
- To develop lambda comparison and expansion operation to build multiple 3-states markov decision models for whole population strings bits auto classification.

Uncertainty decision modelling aims:

- A mixture of discrete and continuous uncertainty distribution is developed.
- Introduce an axiomatic uncertain measure theoretical framework and review framework, the measure theoretical statistical decision theory for preparing the new general uncertainty decision theory developments
- To develop and explore the mathematical foundation, the distribution family, the basic elements underlying uncertainty decision environment.

- To develop and explore a new general uncertainty decision theory framework.
- To apply Bayesian decision method under uncertainty measure theory and created the Bayesian uncertainty decision theory

Differential equation associate regression modelling aims:

- To develop and explore DEAR model use new methods: machine learning.
- Use Lambda algorithm as an optimization tool for DEAR model parameters estimation.

## 1.3 Overview of Thesis

The thesis is divided into five broad chapters: chapter 1 - Introduction, chapter 2 - An Introduction to Theoretical Foundations and Methodology, chapter 3 - Collection of Papers, chapter 4 - Discussion, and chapter 5 - Conclusion.

In chapter 1, The thesis topic motivation is examined and the need to contribute research efforts to a new Statistical Machine Learning Algorithm – DEAR λ- algorithm is described, the aims and objectives for the thesis are stated, and a general overview of the thesis is given.

In chapter 2, background to the subject of SML and the mathematical and statistical theoretical foundations and methodology used in the thesis are discussed.

In chapter 3, seven journal papers were written in contribution to the DEAR λ-algorithm, a development of Statistical Machine Learning Algorithm, of which six are already published and one is under review process. Figure 1.3.1 provides an overview of the thesis papers within the research process.

In chapter 4, the papers are actively examined and criticized, and an overall view of the current development in the research is discussed, particularly the stochastic behaviour of λ-global optimization searching scheme is detail-explained, and the future developments and new directions for further research are outlined.

In chapter 5, a summary of the thesis is given, stating the contribution of this thesis to SML literature, and examine the achievement of aims stated in chapter 1.

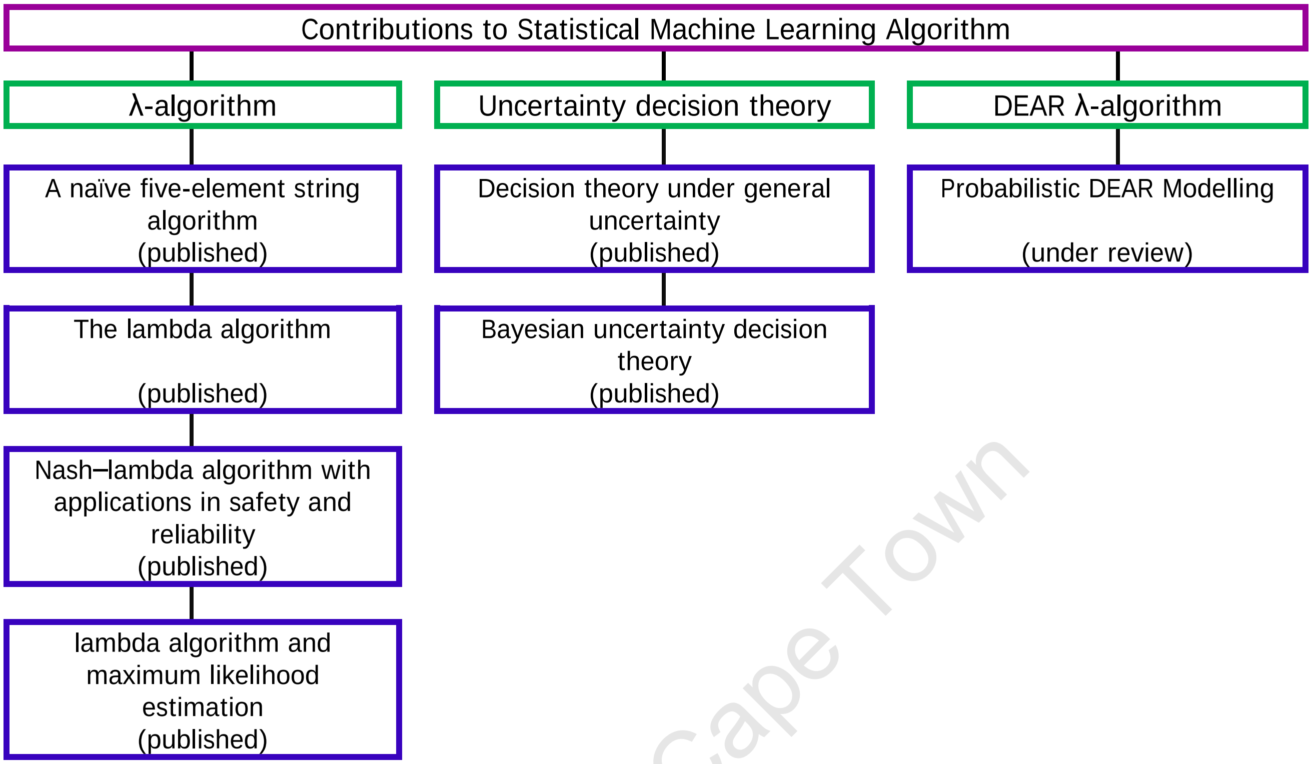



Figure 1.3.1 Overview of the Thesis

# Declaration

I, YanHong Cui, hereby,

1. grant the University of Cape Town free licence to reproduce the above thesis in whole or in part, for the purpose of research;
2. declare that:
   i. the above thesis is my own unaided work, both in concept and execution,
   ii. neither the substance nor any part of the above thesis has been submitted in the past, or is being, or is to be submitted for a degree at this University or at any other university.

I am now presenting the thesis for examination for the degree of Doctor of Philosophy.

YanHong Cui

# Abstract

In this thesis, the issue of model selection compositional elements of the *Statistical Machine Learning* (SML) algorithm is addressed, and also it addresses the issue of optimization scheme, which is called λ-global optimization scheme to make the *Differential Equation Associated Regression* (DEAR) modelling as one of the fastest and most efficient and accurate SML algorithm, which is the new merging frontier of modern computer science and statistics. To support the SML research ideas, seven journal papers, in which six journal papers was already published, were written.

Therefore, the new global optimization scheme (i.e., λ-optimization scheme) is explored, which is reported in four of the journal papers. At the same time, the uncertainty decision analysis within uncertain measure theoretical framework is also explored, which is reported in two journal papers. In the seventh paper, it addresses the simplification of DEAR modelling such that the probabilistic DEAR models will become a practical statistical machine learning algorithm. In other words, a DEAR family which is constituted by a random function with a linear difference equation-wise regression as the central tendency and a variance bound specified by Gaussian error analysis theory is developed delicately, in which the prior distribution will be facilitated by a Gaussian process such that the replication of sampling for estimating the weight matrix will be avoided.

To reach the goal of the new SML algorithm including λ-scheme and the model selection is through a multitude of different papers. The thesis starts with the impreciseness issue of the grey differential equation by engaging investigation of autocovariance structure in the uncertain canonical process, then the uncertain canonical process temporal models, the uncertain canonical process regression models, and the uncertain canonical process DEAR models. From these the uncertain canonical process regression models' research, the thesis reached the formation of the destination of probabilistic DEAR modelling as a conclusion.

# Acknowledgements

My deep thanks to my supervisor, Professor Renkuan Guo, for his invaluable advice and patient guidance in paper writings and publications. Also I must thank my supervisor for using his own research funding to support me attending international conferences for presenting papers. The funding includes South African National Research Foundation Grant No. FA2006042800024, IFR2010042200062, IFR2011040400096, and the Science Faculty URC Funding, and the Postgraduate Fellowship 2011, University of Cape Town.

I would like to thank Associate Professor Christien Thiart and Professor Tim Dunne for offering many advices in my English improvements.

Deep-hearted thanks to my parents Wei Cui and Chungqi Zhang, and wife Ge Wang, for their financial supports and constant encouragements.

A special thanks to all my friends, particularly, Mr. Bin Li and Aunt Yuxia Sun for their care during my PhD study, without their help I can't make today's research achievements.

A special thank to Dr Danni Guo for providing me the thesis template and editing my entire thesis in detail.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 2. Mathematical Background

In this chapter, mathematical and statistical theories used in the thesis are discussed in detail from eight aspects: uncertain measure and uncertain calculus, autocovariance in an uncertain canonical process, statistical decision theory, game theory and Nash equilibrium, weighted least squares regression, Gaussian process regression, maximum likelihood estimation, and Genetic Algorithm (GA).

## 2.1 Uncertain measure and uncertain calculus



Impreciseness is an essential problem we always deal with when we collect some information ready for mathematical models and analysis. Different Impreciseness events in mathematical field may deserve different measure theory to describe with.

In probability theory, Kolmogorov in (1950) invents three axioms (also named as Kolmogorov axioms) to define probability measure theory. Professor Liu proposed an axiomatic uncertain measure theory (2007, 2010, 2011), and opened a new statistical research direction. Liu's uncertain measure is sub-$\sigma$-additive and less restrictive than the ($\sigma$-additive) probability measure. By virtue of its less restrictive nature, Liu's mathematical theory can support the analysis and modeling under more general uncertainty data. Due to our applications, many papers contain the aspects of uncertain calculus. My review is based on Guo, et al. (2011).

Uncertain measure is an axiomatically defined set function mapping from a $\sigma$-algebra of a given space (set) to the unit interval [0,1], which provides a measuring grade system of an uncertain phenomenon and facilitates the formal definition of an uncertain variable.

Let $\Xi$ be a nonempty set (space), and $\mathcal{A}$ the $\sigma$-algebra on $\Xi$. Each element, let us say, $A \subset \Xi, A \in \mathcal{A}$ is called an uncertain event. A number denoted as $\mathcal{M}\{A\}$, $0 \le \mathcal{M}\{A\} \le 1$, is assigned to event $A \in \mathcal{A}$, which indicates the uncertain measuring grade with which event $A \in \mathcal{A}$ occurs. The normal set function $\mathcal{M}\{A\}$ satisfies following axioms given by Liu (2011):

**Axiom 1:** (Normality) $\mathcal{M}\{\Xi\} = 1$.

**Axiom 2:** (Self-Duality) $\mathcal{M}\{\cdot\}$ is self-dual, i.e., for any $A \in \mathcal{A}$, $\mathcal{M}\{A\} + \mathcal{M}\{A^c\} = 1$.

**Axiom 3:** ($\sigma$-Subadditivity) $\mathcal{M}\left\{\bigcup_{i=1}^{\infty} A_i\right\} \le \sum_{i=1}^{\infty} \mathcal{M}\{A_i\}$ for any countable event sequence $\{A_i\} \subset \mathcal{A}$.

**Definition 1:** (Liu, 2007, 2010, 2011) Any set function $\mathcal{M}: \mathcal{A} \to \; 0,1$ satisfies *Axioms 1-3* is called an uncertain measure. The triple $\; \Xi, \mathcal{A}, \mathcal{M} \;$ is called the uncertainty measure space.

We compare Liu's uncertain measure with Kolmorogov's probability measure in Table 2.1.1.

Table 2.1.1 Two different measure axioms comparison

| | Probability measure | Uncertain measure |
|---|---|---|
| Set | $\Omega$ | $\Xi$ |
| Set class | $\sigma$-algebra, $\mathcal{F}$ of $\Xi$ | $\sigma$-algebra, $\mathcal{A}$ of $\Xi$ |
| Axioms | 1. (Normality Axiom)<br>$\Pr\{\Omega\}=1$<br>2. (Numeric bound Axiom)<br>$0 \le P\{A\} \le 1$, for $\forall A \in \mathcal{F}$<br>3. (Countable additivity Axiom)<br>$\forall \{A_n\}_{n=1}^{+\infty} \subset \mathcal{F},\ A_i \cap A_j = \varnothing,$<br>$P\left\{\bigcup_{n=1}^{+\infty} A_n\right\} = \sum_{n=1}^{+\infty} P\{A_n\}$<br>$i \ne j,\ \forall i, j = 1,2,\cdots$ | 1.(Normality Axiom)<br>$\mathcal{M}\{\Xi\}=1$<br>2.( Self-Duality Axiom)<br>$\mathcal{M}\{A\}+\mathcal{M}\{A^c\}=1,\ \forall A \in \mathcal{A}$<br>3.Countable Subadditivity Axiom)<br>$\forall \{A_n\}_{n=1}^{+\infty} \subset \mathcal{A},$<br>$\mathcal{M}\left\{\bigcup_{n=1}^{+\infty} A_n\right\} \le \sum_{n=1}^{+\infty} \mathcal{M}\{A_n\}$ |
| Set Mapping | $P:\mathcal{F} \to [0,1]$ | $\mathcal{M}:\mathcal{A} \to [0,1]$ |
| Product Measure | | (Product measure Axiom)<br>$\mathcal{M}\left\{\prod_{m=1}^{d} A_m\right\} = \min_{1\le m\le d} \mathcal{M}\{A_m\}$ ,i.e.,<br>for each event $A \in \mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_m$<br>$\mathcal{M}\{A\} = \begin{cases} \sup\limits_{A_1\times\cdots\times A_d \subset A} \min\limits_{1\le m\le d}\{\mathcal{M}\{A_m\}\} & \text{if } \sup\limits_{A_1\times\cdots\times A_d \subset A} \min\limits_{1\le m\le d}\{\mathcal{M}\{A_m\}\} > 0.5 \\ 1-\sup\limits_{A_1\times\cdots\times A_d \subset A^c} \min\limits_{1\le m\le d}\{\mathcal{M}\{A_m\}\} & \text{if } \sup\limits_{A_1\times\cdots\times A_d \subset A^c} \min\limits_{1\le m\le d}\{\mathcal{M}\{A_m\}\} > 0.5 \\ 0.5 & \text{otherwise} \end{cases}$ |



**Definition 2:** (Liu, 2007, 2010, 2011) An uncertain variable $\xi$ is a measurable mapping, i.e., $\xi:(\Xi,\mathcal{A}) \to (\mathbb{R},\mathcal{B})$, where $\mathcal{B}$ denotes the Borel $\sigma$-algebra on $\mathbb{R}=(-\infty,+\infty)$.

In uncertain variable definition, the measurable mapping is characterized by the membership of the pre-image of event (a Borel set) $B = \; -\infty, r \;$ under the uncertain variable $\xi$ to the $\sigma$-algebra $\mathcal{A}$. In other words,

$$\forall B \in \mathcal{B}, \{\tau \in \Xi : \xi \in B\} \in \mathcal{A} \tag{1}$$

The measurability of uncertain variable $\xi$ definitely induces a measure on the measurable space $(\mathbb{R},\mathcal{B})$.

**Definition 3:** (Liu, 2007, 2010, 2011) The uncertain distribution $\Psi:\mathbb{R}\to[0,1]$ of an uncertain variable $\xi$ on $\Xi,\mathcal{A},\lambda$ is

$$\Psi(x)=\lambda\left\{\tau\in\Xi\middle|\xi(\tau)\le x\right\} \tag{2}$$

In probabilistic modeling, the distribution function of a random variable plays the same role as probability measure which assigns the measure grade to the event characterized by the set in which random variable takes values, i.e., the probability measure and distribution of an random variable are equivalent. But in Liu's uncertainty theory (2007, 2010, 2011), only the uncertain measure governing the uncertainty variable can fully assigning the measure grade to the event characterized by the set in which random variable takes values. The impact of such a difference causes many mathematical characteristics, for example, variance of an uncertainty variable cannot be derived accurately like the variance of a random variable in probability theory, rather, only an upper bound can be specified and stipulated as the variance of an uncertainty variable.

**Definition 4:** (Liu, 2007, 2010, 2011) Let $\xi$ be a uncertainty variable on an uncertainty measure space $\Xi,\mathcal{A},\lambda$. The expectation $\xi$ is defined by

$$\mathrm{E}[\xi]=\int_{0}^{+\infty}\lambda\{\xi\ge s\}\,ds-\int_{-\infty}^{0}\lambda\{\xi\le s\}\,ds \tag{3}$$

provided that one of the two integrals exists at least.

**Definition 5:** The upper bound of $\mathrm{E}\left[(\xi-\mu)^2\right]$ is defined as the variance of $\xi$, denoted by $V[\xi]$, if $\mu=\mathrm{E}[\xi]$ and $\mathrm{E}\left[(\xi-\mu)^2\right]$ exist and are finite.

**Theorem 6:** (Liu, 2007, 2010, 2011) Let $\xi$ be a uncertainty variable on an uncertainty measure space $(\Xi,\mathfrak{A}(\Xi),\lambda)$.with uncertainty distribution function and $\Psi_\xi$ a finite expectation $\mu$. Then

$$V[\xi]\triangleq\sup\left\{\mathrm{E}\left[(\xi-\mu)^2\right]\right\}=2\int_{0}^{+\infty}(r-\mu)\left(1-\Psi_\xi(r)+\Psi_\xi(2\mu-r)\right)dr. \tag{4}$$

Similar to probability theory, uncertain process can be defined as a family of uncertainty variables denoted as $\{\xi_t,t\in\mathbb{T}\}$, which take values from the space $\mathbb{S}$ and indexed by common index space $\mathbb{T}$. The most important uncertain process is the uncertain canonical process, which plays the roles similar to that of Brownian motion process in probability theory.

**Definition 7:** (Liu, 2007, 2010, 2011) Let $\{C_t, t \geq 0\}$ be a standard uncertain process.

(1) $C_0 = 0$ and all the trajectories of realizations are Lipschitz-continuous;

(2) $\{C_t, t \geq 0\}$ has stationary and independent increments;

(3) every increment $C_{t+s} - C_s$ is a normal uncertainty variable with expected value 0 and variance $t^2$, i.e., the uncertainty distribution of $C_{t+s} - C_s$ is

$$\Psi_{C_{t+s}-C_s}(z) = \left(1 + \exp\left(-\frac{\pi z}{\sqrt{3}t}\right)\right)^{-1} \tag{5}$$

Then, $\{C_t, t \geq 0\}$ is called an uncertain canonical process.

**Remark 8:** Definition 2.7 is a modified version at item (1) of Liu's definition, because in Liu's uncertainty theory, term "sample" and thus term "sample paths" are undefined (Liu, 2007, 2010, 2011). Term "realizations" is more suitable here although in probability theory term "sample" and term "realizations" are synonym almost.

**Definition 9:** (Liu, 2007, 2010, 2011) Suppose $\{C_t, t \geq 0\}$ is an uncertain canonical process, and $f$ and $g$ are some given functions, then

$$d\xi_t = f(t, \xi_t)dt + g(t, \xi_t)dC_t \tag{6}$$

is called an uncertain differential equation. A solution to the uncertain differential equation is the uncertain process $\{\xi_t, t \geq 0\}$ satisfying Eq. (6) for any $t > 0$.

**Remark 10:** Since $dC_t$ and $d\xi_t$ are only meaningful under the umbrella of uncertain integral, i.e., the an uncertain differential equation is an alternative representation of

$$\xi_t = \xi_0 + \int_0^t f(s, \xi_s)ds + \int_0^t g(s, \xi_s)dC_s \tag{7}$$

The uncertain process $\{\xi_t, t \geq 0\}$, whose member $\xi_t$ is the solution to the uncertain differential equation in Eq.(7) or Eq.(8) is driven by the uncertain canonical process $\{C_t, t \geq 0\}$ via the uncertain integral $\int_0^t g(s, \xi_s)dC_s$. Therefore $\{\xi_t, t \geq 0\}$ is called an uncertain integral driven process.

Liu has introduced an important class of uncertain integral $\int_0^t \phi(t)dC_s$ ( Liu, 2007, 2010, 2011), which is uncertain normal variable for any given $t \geq 0$, i.e.,

$$\Psi_{\int_0^t \phi(u)dC_u}(z) = \left(1+\exp\left(-\frac{\pi z}{\sqrt{3}\int_0^t |\phi(u)|\,du}\right)\right)^{-1} \tag{8}$$

**Definition 11:** (Guo et al., 2011) Let $\Phi(t)=\int_0^t \phi(u)dC_u$, where $\{C_t, t \geq 0\}$ is a standard uncertain canonical process and the integrand function $\phi(\cdot)$ of the uncertain integral is a deterministic positive real-valued function of $t$. Then uncertain process $\{\Phi_t, t \geq 0\}$ is simply called as $\Phi$ process.

**Lemma 12:** (Guo et al., 2011) A $\Phi$ process is an uncertain integral driven process, whose member indexed by $t \geq 0$ is a uncertain normal variable.

**Theorem 13:** (Guo et al., 2011) Let $\{\Phi_t, t \geq 0\}$ be a $\Phi$ process. If the integrand function $\phi(\cdot)$ of the uncertain integral is positive and integrable, then the increment of the $\Phi$ process $\{\Phi_t, t \geq 0\}$ is non-stationary uncertain process. If $\int_s^{t+s} \phi(u)du = f(t)$, then the increments of $\Phi$ process $\{\Phi_t, t \geq 0\}$ is stationary.

**Theorem 14:** (Guo et al., 2011) A $\Phi$ process $\{\Phi_t, t \geq 0\}$ has independent increments, i.e., for $0 < t_1 < t_2 < \cdots < t_n < t$, $\Phi_{t_1} - \Phi_0$, $\Phi_{t_2} - \Phi_{t_1}, \cdots, \Phi_{t_n} - \Phi_{t_{n-1}}$ are mutually independent.

An important $\Phi$ process is the fractional uncertain canonical process, given by Liu (2011), $\{F_t, t \geq 0\}$, where $\phi(s) = (t-s)^{-\alpha}$, $\alpha \in (0,1)$, i.e.,

$$F_t = \int_0^t (t-u)^{-\alpha}\, dC_u,\ \alpha \in (0,1) \tag{9}$$

The parameter $\alpha$ is used to control the autocovariance fluctuations. For $\forall t > 0$

$$\Psi_{F_t}(z) = \left(1+\exp\left(-\frac{\pi z}{\sqrt{3}\left(t^{1-\alpha}/(1-\alpha)\right)}\right)\right)^{-1}, \alpha \in (0,1) \tag{10}$$

It is noticed that

$$\int_s^{t+s} (v-u)^{-\alpha}\, du = \frac{1}{1-\alpha} t^{1-\alpha},\ \alpha \in (0,1) \tag{11}$$

because

$$\int_s^{v} (v-u)^{-\alpha}\, du = \frac{1}{1-\alpha} (v-s)^{1-\alpha} \tag{12}$$

Therefore a fractional uncertain canonical process has stationary increments.

## 2.2 Autocovariance in uncertain canonical process

In the thesis, the terminology “autocovariance” is used to serve the intention of emphasizing that the concept “autocovariance” characterizes the linear association between two members within an uncertain canonical process.

Liu (2011) has not defined the uncertain covariance between two uncertainty variables yet. During the investigation of the uncertainty linear regression, Guo et al. (2010) discovered the importance of the uncertain covariance concept. Hence, I simply list the treatments as follows.

Different from probability distribution, which fully specifies every measurable event, the uncertainty distribution can only characterize event $\{\xi \le z\}$ or the event $\{\xi > z\}$ for any $z \in \mathbb{R}$ . In uncertainty theory, only uncertain measure can fully specify any measurable event.

**Definition 15:** (Liu, 2007, 2010, 2011) The uncertainty variables $\xi_1, \xi_2, \cdots, \xi_n$ on $(\Xi, \mathcal{A}, \text{ƛ})$ are said to be independent if

$$\text{ƛ}\left\{\bigcap_{i=1}^{n}\{\xi_i \in B_i\}\right\} = \min_{1\le i\le n} \text{ƛ}\{\xi_i \in B_i\} \tag{13}$$

for any Borel sets $B_1, B_2, \cdots, B_n \in \mathcal{B}$ .

**Theorem 16:** (Liu, 2007, 2010, 2011) Let $\Psi_{\xi_1}, \Psi_{\xi_2}, \cdots, \Psi_{\xi_n}$ be uncertainty distributions for the uncertainty variables $\xi_1, \xi_2, \cdots, \xi_n$ on $(\Xi, \mathcal{A}, \text{ƛ})$ respectively. Let $\Psi_{(\xi_1, \xi_2, \cdots, \xi_n)}$ be the joint distribution of uncertainty vector $(\xi_1, \xi_2, \cdots, \xi_n)$. If $\xi_1, \xi_2, \cdots, \xi_n$ are independent, then

$$\Psi_{(\xi_1, \xi_2, \cdots, \xi_n)}(x_1, x_2, \cdots, x_n) = \min_{1\le i\le n} \Psi_{\xi_i}(x_i) \tag{14}$$

for any real numbers $x_1, x_2, \cdots, x_n \in \mathbb{R}$.

Let $\Psi_{\xi_1}, \Psi_{\xi_2}, \cdots, \Psi_{\xi_n}$ be uncertainty distributions for the uncertainty variables $\xi_1, \xi_2, \cdots, \xi_n$ on $(\Xi, \mathcal{A}, \text{ƛ})$ respectively. Let $\Psi_{(\xi_1, \xi_2, \cdots, \xi_n)}$ be the joint distribution of uncertainty vector $(\xi_1, \xi_2, \cdots, \xi_n)$. Assuming that $\xi_i$ and $\xi_j$ two arbitrary pair of uncertainty variables within the uncertainty vector $(\xi_1, \xi_2, \cdots, \xi_n)$ which have finite expectations $\mu_i$ and $\mu_j$ respectively. Denote $\Psi_{(\xi_i, \xi_j)}$ bivariate uncertainty distribution function.

**Definition 17:** (Guo et al., 2010) The bivariate uncertainty distribution function $\Psi_{(\xi_i, \xi_j)}$ is given by

$$\Psi_{(\xi_i,\xi_j)}(z_i,z_j)=\sup_{(z_1,\cdots z_k,\cdots,z_n)\in\mathbb{R}^{n-2},\ z_k\neq z_i,z_j}\Psi_{(\xi_1,\xi_2,\cdots,\xi_n)}(z_1,z_2,\cdots,z_n). \tag{15}$$

**Definition 18:** (Guo et al., 2010) Let $\eta_{ij}=(\xi_i-\mu_i)(\xi_j-\mu_j)$ be product of centered uncertainty variables $\xi_i$ and $\xi_j$. The uncertainty distribution of $\eta_{ij}$ is defined by

$$\Psi_{\eta_{ij}}(y)=\sup_{z_1z_2=y}\Psi_{(\xi_i,\xi_j)}(z_i,z_j),\ \forall(z_i,z_j)\in\mathbb{R}^2 \tag{16}$$

**Definition 19:** (Guo et al., 2010) Let $\eta_{ij}=(\xi_i-\mu_i)(\xi_j-\mu_j)$ be product of centered uncertainty variables $\xi_i$ and $\xi_j$. The expectation of centered product $\eta_{ij}$ is called the covariance between uncertainty variables $\xi_i$ and $\xi_j$, that is

$$\sigma_{ij}\triangleq\mathrm{E}[\eta_{ij}]=\int_0^{+\infty}\left(1-\Psi_{\eta_{ij}}(r)\right)dr-\int_{-\infty}^{0}\Psi_{\eta_{ij}}(r)dr \tag{17}$$

**Theorem 20:** (Guo et al., 2011) Let $\xi_1,\xi_2,\cdots,\xi_n$ be independent uncertainty variables on $(\Xi,\mathcal{A},\lambda)$. Then

$$\Psi_{\eta_{ij}}(y)=\sup_{z_1z_2=y}\left(\Psi_{\xi_1}(z_1)\wedge\Psi_{\xi_2}(z_2)\right),\ \forall(z_i,z_j)\in\mathbb{R}^2 \tag{18}$$

**Remark 21:** Different from independent random variables $X_1$ and $X_2$, whose $\mathrm{E}\left[(X_1-\mathrm{E}[X_1])(X_2-\mathrm{E}[X_2])\right]=0.$ In uncertainty theory, independent $\xi_1,\xi_2,\cdots,\xi_n$ do not imply $\eta_{ij}=0$.

**Definition 22:** The matrix is called variance-covariance matrix $(\sigma_{ij})_{n\times n}$, denoted by

$$\Sigma\triangleq(\sigma_{ij})_{n\times n}=\begin{bmatrix} V[\xi_1] & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & V[\xi_2] & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & V[\xi_n] \end{bmatrix} \tag{19}$$

**Theorem 23:** The variance-covariance matrix is symmetric. Furthermore, $\Sigma=PDP'$ such that $P'P=PP'=I$ and $D=\mathrm{diag}_n(\lambda_i)$ where $\lambda_i$ is the $i^{th}$ largest eigenvalue and $P=(\underline{e}_1,\underline{e}_2,\cdots,\underline{e}_n)$ with $\underline{e}_i$ the $i^{th}$ eigenvector corresponding to $\lambda_i$, $i=1,2,\cdots,n.$

**Remark 24:** The formation of uncertain variance-covariance does touch the detailed functional form of multivariate uncertainty joint distribution. For practical applications, the form of multivariate uncertainty joint distribution is not necessarily available, but the paired uncertainty bivariate distribution must be given.

We use term "autocovariance" indicates those covariances are linear associations among the members (which are uncertainty variables anyway) of an uncertain process, indexed by $t \in T = [0, +\infty)$. The index set is time, therefore those indexed members denoted by $\{\Phi_{t_1}, \Phi_{t_2}, \cdots, \Phi_{t_n}\}$, behave similar to time-series. That is the reason why the term "autocorrelation" is used in this paper.

**Theorem 25:** (Guo et al., 2011) Let $\{\Phi_t, t \geq 0\}$ be a $\Phi$ process. If the integrand function $\phi(\cdot)$ of the uncertain integral is positive and integrable. Then for $\forall s < t$, the autocovariance $\sigma_{s,t} = \mathrm{E}[\Phi_s \Phi_t]$ and autocorrelation $\rho_{s,t}$ of the $\Phi$ process $\{\Phi_t, t \geq 0\}$ are

$$\sigma_{s,t} = \int_0^s \phi(u)\,du \int_0^t \phi(u)\,du - \delta_{s,t-s} \tag{20}$$

and

$$\rho_{s,t} = 1 - \frac{\delta_{s,t-s}}{\int_0^s \phi(u)\,du \int_0^t \phi(u)\,du}, \quad (s < t) \tag{21}$$

where

$$\delta_{s,t-s} = \int_{-\infty}^{0} \sup_{y^2+4r \geq 0} \left( \left( \Psi_{\Phi_s}\left( \frac{y+\sqrt{y^2+4r}}{2} \right) - \Psi_{\Phi_s}\left( \frac{y-\sqrt{y^2+4r}}{2} \right) \right) \wedge \Psi_{\Phi_t-\Phi_s}(y) \right) dr \tag{22}$$

**Proof:** By definition, the autocovariance is

$$\begin{aligned} \sigma_{s,t} &= \sigma_{\Phi_s,\Phi_t} \triangleq \mathrm{E}[\Phi_s \Phi_t] = \mathrm{E}\left[\Phi_s\left(\Phi_s + (\Phi_t - \Phi_s)\right)\right] \\ &= \mathrm{E}\left[\Phi_s^2 + \Phi_s(\Phi_t - \Phi_s)\right] \end{aligned} \tag{23}$$

Notice that the increment $\Phi_t - \Phi_s$ is independent of $\Phi_s$ for any two time points *s* and *t* as long as *s* < *t*. The marginal uncertainty distribution for $\Phi_s$ is

$$\Psi_{\Phi_s}(u) = \left( 1 + \exp\left( -\frac{\pi u}{\sqrt{3}\int_0^s \phi(u)\,du} \right) \right)^{-1} \tag{24}$$

and the marginal uncertainty distribution for $\Phi_t - \Phi_s$ is

$$\Psi_{C_t - C_s}(v) = \left( 1 + \exp\left( -\frac{\pi v}{\sqrt{3}\int_s^t \phi(u)\,du} \right) \right)^{-1} \tag{25}$$

The joint uncertainty distribution for the joint uncertain variables $\Phi_s$ and $\Phi_t - \Phi_s$ is

$$\begin{aligned}&\Psi_{\Phi_s,\Phi_t-\Phi_s}(u,v) = \tilde{\lambda}\{\Phi_s \le u, \Phi_t - \Phi_s \le v\} \\ &= \tilde{\lambda}\{\Phi_s \le u\} \wedge \tilde{\lambda}\{\Phi_t - \Phi_s \le v\} = \Psi_{\Phi_s}(u) \wedge \Psi_{\Phi_t-\Phi_s}(v)\end{aligned} \tag{26}$$

Thus, according to the definition of expected value, we have

$$\begin{aligned}&\sigma_{s,t} = \mathrm{E}[\Phi_s\Phi_t] = \mathrm{E}\left[\Phi_s^2 + \Phi_s(\Phi_t - \Phi_s)\right] \\ &= \int_0^{+\infty} \tilde{\lambda}\left\{\Phi_s^2 + \Phi_s(\Phi_t - \Phi_s) \ge r\right\} dr \\ &- \int_{-\infty}^{0} \tilde{\lambda}\left\{\Phi_s^2 + \Phi_s(\Phi_t - \Phi_s) \le r\right\} dr\end{aligned} \tag{27}$$

For the first integrand in Eq. (27), the upper bound of it is

$$\begin{aligned}&\tilde{\lambda}\left\{\Phi_s^2 + \Phi_s(\Phi_t - \Phi_s) \ge r\right\} \\ &= \tilde{\lambda}\left\{\left(\bigcup_{\substack{x>0 \\ x^2+xy\ge r}} (\Phi_s \ge x, \Phi_t - \Phi_s \ge y)\right) \cup \left(\bigcup_{\substack{x<0 \\ x^2+xy\ge r}} (\Phi_s \le x, \Phi_t - \Phi_s \le y)\right)\right\} \\ &\le \tilde{\lambda}\left\{\bigcup_{\substack{x>0 \\ x^2+xy\ge r}} (\Phi_s \ge x, \Phi_t - \Phi_s \ge y)\right\} + \tilde{\lambda}\left\{\bigcup_{\substack{x<0 \\ x^2+xy\ge r}} (\Phi_s \le x, \Phi_t - \Phi_s \le y)\right\} \\ &= 1 - \sup_{\substack{x>0 \\ x^2+xy=r}} \left(\Psi_{\Phi_s}(x) \wedge \Psi_{\Phi_t-\Phi_s}(y)\right) + \sup_{\substack{x<0 \\ x^2+xy=r}} \left(\Psi_{\Phi_s}(x) \wedge \Psi_{\Phi_t-\Phi_s}(y)\right)\end{aligned} \tag{28}$$

Furthermore,

$$\Psi_{\Phi_s}(x) \wedge \Psi_{\Phi_t-\Phi_s}(y) = \left(1 + \exp\left(-\frac{\pi}{\sqrt{3}}\left(\frac{x}{\int_0^s \phi(u)\,du} \wedge \frac{y}{\int_s^t \phi(u)\,du}\right)\right)\right)^{-1} \tag{29}$$

Thus, it is easy to verify that with constraint $x^2 + xy = r$, function $\Psi_{\Phi_s}(x) \wedge \Psi_{\Phi_t-\Phi_s}(y)$ reaches its maximum if and only if

$$\frac{x}{\int_0^s \phi(u)\,du} = \frac{y}{\int_s^t \phi(u)\,du}, \tag{30}$$

i.e.

$$\begin{cases} x = \sqrt{\dfrac{r\int_0^s \phi(u)\,du}{\int_0^t \phi(u)\,du}} \\ y = \dfrac{\sqrt{r}\int_s^t \phi(u)\,du}{\sqrt{\int_0^s \phi(u)\,du \int_0^t \phi(u)\,du}} \end{cases} \tag{31}$$

thus, we obtain an interesting result for the first term in Eq.(27):

$$
\begin{aligned}
&\int_{0}^{+\infty} ƛ\left\{\Phi_s{}^2+\Phi_s\left(\Phi_t-\Phi_s\right)\geq r\right\}dr \\
&\leq \int_{0}^{+\infty}\left(1-\left(\Psi_{\Phi_s}\left(\sqrt{\frac{r\int_0^s\phi(u)du}{\int_0^t\phi(u)du}}\right)\right)+\Psi_{\Phi_s}\left(-\sqrt{\frac{r\int_0^s\phi(u)du}{\int_0^t\phi(u)du}}\right)\right)dr \\
&=\frac{\int_0^t\phi(u)du}{\int_0^s\phi(u)du}V\left[\Phi_s\right]=\int_0^s\phi(u)du\int_0^t\phi(u)du
\end{aligned} \tag{32}
$$

As to the second term in Eq.(27), we notice that

$$
\begin{aligned}
&\int_{-\infty}^{0} ƛ\left\{\Phi_s{}^2+\Phi_s\left(\Phi_t-\Phi_s\right)\leq r\right\}dr \\
&\geq \int_{-\infty}^{0} ƛ\left\{\bigcup_{\substack{x_1<x_2\leq 0\\ x_1{}^2+x_1y=r\\ x_2{}^2+x_2y=r}}\left(x_1\leq\Phi_s\leq x_2,\Phi_t-\Phi_s\geq y\right)\right\}dr \\
&\geq \int_{-\infty}^{0}\sup_{y^2+4r\geq 0}\left(\left(\Psi_{\Phi_s}\left(\frac{y+\sqrt{y^2+4r}}{2}\right)-\Psi_{\Phi_s}\left(\frac{y-\sqrt{y^2+4r}}{2}\right)\right)\wedge\Psi_{\Phi_t-\Phi_s}(y)\right)dr
\end{aligned} \tag{33}
$$

Combining the arguments from Eq.(26) to Eq.(33), we have

$$
\begin{aligned}
&\mathrm{E}\left[\Phi_s{}^2+\Phi_s\left(\Phi_t-\Phi_s\right)\right] \\
&\leq st-\int_{-\infty}^{0}\sup_{y^2+4r\geq 0}\left(\left(\Psi_{\Phi_s}\left(\frac{y+\sqrt{y^2+4r}}{2}\right)-\Psi_{\Phi_s}\left(\frac{y-\sqrt{y^2+4r}}{2}\right)\right)\wedge\Psi_{\Phi_t-\Phi_s}(y)\right)dr
\end{aligned} \tag{34}
$$

As a matter of fact, we cannot calculate the exact value of an autocovariance with an uncertain distribution which only carry a part of information of the uncertain measure. For convenience, we stipulate that the autocovariance is defined by its upper bound, i.e.,

$$\sigma_{s,t}=\mathrm{E}\left[\Phi_s{}^2+\Phi_s\left(\Phi_t-\Phi_s\right)\right]=\int_0^s\phi(u)du\int_0^t\phi(u)du-\delta_{s,t-s} \tag{35}$$

where

$$\delta_{s,t-s}=\int_{-\infty}^{0}\sup_{y^2+4r\geq 0}\left(\left(\Psi_{\Phi_s}\left(\frac{y+\sqrt{y^2+4r}}{2}\right)-\Psi_{\Phi_s}\left(\frac{y-\sqrt{y^2+4r}}{2}\right)\right)\wedge\Psi_{\Phi_t-\Phi_s}(y)\right)dr \tag{36}$$

Accordingly, the autocovariance and autocorrelation are

$$\sigma_{s,t}=\int_0^s\phi(u)du\int_0^t\phi(u)du-\delta_{s,t-s} \tag{37}$$

and

$$\rho_{s,t}=\frac{\sigma_{\Phi_s\Phi_t}}{\sqrt{V\left[\Phi_s\right]}\sqrt{V\left[\Phi_t\right]}}=1-\frac{\delta_{s,t-s}}{\int_0^s\phi(u)du\int_0^t\phi(u)du} \tag{38}$$

respectively.

**Corollary 26:** The $\delta_{s,t-s}$ function tends to zero if *t* tends to *s*, which implies that

$\sigma_{s,t} = \int_0^s \phi(u)\,du \int_0^t \phi(u)\,du - \delta_{s,t-s}$ is a continuous function and $\lim_{t\to s} \sigma_{s,t} = \left(\int_0^s \phi(u)\,du\right)^2 = \sigma^2_{\Phi_s}$

For practical applications, the time index is often taking integer value. Therefore, we state a corollary for dealing with such a case.

**Corollary 27:** The autocovariance for the integer-value indexed numerical

$\delta_{i,j-i}$, $i, j = 1, 2, \cdots, N$ , , $i < j$ ,

$$\sigma_{i,j} = \sigma_{\Phi_i \Phi_j} = \int_0^i \phi(u)\,du \int_0^j \phi(u)\,du + \delta_{i,j-i} \tag{39}$$

## 2.3 Basic elements of statistical decision theory

Statistical modeling and study of stochastic behavior of optimization scheme are statistical decision problems. Therefore, it is necessary to have knowledge of the three basic elements, i.e., the *state, action, and loss* in the statistical decision theory (Chen, 1981; DeGroot, 1970; Ferguson, 1967; Guo, 2011, Lee, 1989), which are still the essential elements in the new general uncertainty decision theory.

Firstly, in statistical decision theory, the state, termed “state of nature” is regarded as objectively in existence, at least in some consensus sense.

Secondly, the connotations of “action” in the decision theory is virtual, in which some elements are of a precautionary nature and do not correspond to any specific state element. The mapping is of multiple states to multiple action nature. However, the inclusion of virtual action elements is extremely important, because the top decision maker does not need to deal with routine decisions of day-to-day operations but with the extreme event or the most important event decision.

Thirdly, the loss is a selection, which minimizes the loss function $l(\theta, a)$ of an action $a$ from action space $\mathbb{A}$ for given state $\theta$ in the state space $\Theta$. In the statistical decision theory, an action is made in terms of observational data, denoted as $x$, which is described by a probability distribution $F(x|\theta)$. Based on data $x$ (i.e., $F(x|\theta)$), a decision is actually a mapping from data space $\mathbb{D}$ into action space $\mathbb{A}$. In other words,

$$a : \mathbb{D} \to \mathbb{A} \tag{40}$$

which can be expressed by

$$a = d\left(x\right) \tag{41}$$

The loss $l\left(\theta, d\left(x\right)\right)$ is measurable on the joint uncertainty space.

**Definition 28:** The expected value of the loss with respect to the distribution of uncertain data $x$.

$$R\ \theta, d\ = \mathrm{E}_{\theta}\left[l\ \theta, d\ x\ \right] \tag{42}$$

is called a risk function.

The distribution of uncertain data $z$ depends on state $\theta$, because the dependence of $R\left(\theta, d\right)$ on $\theta$ enters explicitly from $l\left(\theta, a\right)$ and also through the state $\theta$ in the distribution function $F\left(x \mid \theta\right)$ for $x$. Therefore the probability distribution of the data determines the fundamental characteristics of observational data oriented statistical decision theory.

Let us consider the statistical decision problem for a given probability distribution. Assume a state space $\Theta = \mathbb{R}$, and a continuous action space $\mathbb{A} = \mathbb{R}$, and the quadratic loss function is defined by

$$l\left(\theta, a\right) = w\left(\theta\right)\left(\theta - a\right)^{2} \tag{43}$$

**Definition 29:** (Bayes loss) Given a continuous state space $\Theta$, the uncertainty variable $\theta$ is defined on uncertain space $\left(\Theta, \mathcal{B}_{\Theta}, P_{\theta}\right)$, where $P_{\theta}\left(\cdot\right)$ is a probability measure. The probability distribution $F_{\theta}\left(\cdot\right)$ is defined on $\left(\Theta, \mathcal{B}_{\Theta}\right)$. Then the average of loss with respect to state space for a given action $a \in \mathbb{A}$, is called the Bayes loss for a given action $a$:

$$B\left(a\right) = \mathrm{E}\left[l\left(\theta, a\right)\right] = \int_{\Theta} l\left(y, a\right) dF_{\theta}\left(y\right) \tag{44}$$

**Definition 30:** (Bayes rule) A Bayes decision rule, denoted as $d^{B}$, is a rule such that the Bayes risk is minimized, i.e.,

$$B\left(d^{B}\right) = \min_{d \in \mathbb{D}}\left\{B\left(d\right)\right\} \tag{45}$$

The basic form of Bayesian decision can be illustrated by following formulation and example. When evidence is denoted by $\underline{x}$, is scarce, an index of the system, denoted by $s$ $(0 \le s \le 1)$, we may use the soft evidence on *s* in the form of a prior density ρ(*s*) in terms of Bayes' theorem to calculate the posterior density of *s*,

$$f(s \mid \underline{x}) = \frac{l(s \mid \underline{x})\rho(s)}{\int_0^1 l(s \mid \underline{x})\rho(s)ds} \tag{46}$$

where $l(s \mid \underline{x}) \triangleq f(\underline{x} \mid s)$ is called the likelihood function, obtained from the joint density of the sample evidence $\underline{x}$ .

For example, *N* electronic devices are under testing until a preset time *T*, by assuming that the failure time of a random individual device follows an exponential distribution with the density:

$$f(t \mid \lambda) = \lambda e^{-\lambda t}, t \geq 0, \lambda > 0. \tag{47}$$

Suppose that in testing period [0,*T*], *x* units failed and the failure times are recorded as $t_1, t_2, \cdots, t_x$ , thus the sample evidence as $\underline{t} = \left(t_1, t_2, \cdots, t_x, T, T, \cdots, T\right)$, and the likelihood function is then given by

$$l(\lambda \mid \underline{t}) = \left\lfloor \prod_{i=1}^{x} \left(\lambda e^{-\lambda t_i}\right) \right\rfloor \left[1 - \left(1 - e^{-\lambda T}\right)\right]^{n-x} = \lambda^x e^{-\lambda \eta}, \tag{48}$$

where $\eta = \left(n - x\right)T + \sum_{i=1}^{x} t_i$ . Then a gamma prior density on the failure rate $\lambda$ , with priori parameters $\alpha$ and $\beta$ , may represent the expert knowledge:

$$\rho(\lambda) = \frac{\beta^\alpha}{\Gamma\left(\alpha\right)} \lambda^{\beta - 1} e^{-\alpha\lambda}, \lambda > 0, \alpha \geq 0, \beta \geq 0. \tag{49}$$

The posterior density of the failure rate $\lambda$ is evaluated in terms of Eq. (50):

$$f(\lambda \mid \underline{x}) = \frac{l(\lambda \mid \underline{x})\rho(\lambda)}{\int_0^1 l(\lambda \mid \underline{x})\rho(\lambda)d\lambda} = \frac{\left(\alpha + \eta\right)^{\beta + x}}{\Gamma\left(\alpha + x\right)} \lambda^{\beta + x + 1} e^{-(\alpha + \eta)\lambda}, \tag{50}$$

which is also a gamma density because the prior density takes the form of the conjugate family. Further decision analysis or inference will be based on the posterior density. For detail, see Guo (2011), Guo et al. (2011).

## 2.4 Game theory and Nash equilibrium

The game theory is an applied mathematical branch dealing with the behaviour in strategic situations, in which an individual's gain in making choices depends on the choices of the individual's competitors. Game theory studies theory on the rational side of social science in broad sense, including human as well as non-human players e.g., computers, animals, and etc. (Wikipedia, 2011).

In *n*-player non-corporative games, the Nash equilibrium is a solution state, in which an individual player knows the strategies of the others and also knows that no one can gain anything by altering any individual strategy unilaterally while the others keep their strategies unchanged. Such a set of strategy choices and the corresponding payoffs constitute a Nash equilibrium (Wikipeidia, 2011).

Let $(S, f)$ be a game with $n$ players, in which $S = S_1 \times S_2 \cdots \times S_n$ is the strategy-profile set with the $i^{th}$ player's strategy set $S_i$, $i = 1, 2, \cdots, n$, and $f = f\left(f_1(x), f_2(x), \cdots, f_n(x)\right)$ is the payoff function. When each individual player decides to choose the strategy $x_i$, then a strategy profile $x = \left(x_1, \cdots, x_n\right)$ is obtained so that the $i^{th}$ player $i$ obtains payoff $f_i(x)$. Let $x_{-i}$ be a strategy profile of all players except for the $i^{th}$ player. Note that the payoff depends on the strategy profile chosen, i.e. on the strategy chosen by player $i$ as well as the strategies chosen by all the remaining players.

**Definition 31:** A strategy profile $x^* \in S$ is Nash equilibrium if no unilateral deviation in strategy by any individual player is profitable for that player, that is

$$\forall i, x_i \in S_i, x_i \neq x_i^* : f_i\left(x_i^*, x_{-i}^*\right) \geq f_i\left(x_i, x_{-i}^*\right). \tag{51}$$

A game can have either a pure-strategy or a mixed-strategy Nash Equilibrium, (in the latter a pure strategy is chosen stochastically with a fixed frequency). Nash proved that if we allow mixed strategies, then every game with a finite number of players in which each player can choose from finitely many pure strategies has at least one Nash equilibrium solution (Wikipedia, 2011].

## 2.5 Weighted least squares linear regression

In statistical linear regression theory (Chen, 1981; Draper and Smith, 1966; Myers, 2000; Rao, 1973), the basic assumptions are:

**Assumption 1:** The model takes a form:

$$y_i = \beta_0 + \beta_1 x_{1,i} + \ldots + \beta_{p,i} x_{p,i} + \varepsilon_i,\ i = 1,2,\cdots,n \tag{52}$$

**Assumption 2:** The error terms satisfy several conditions:

(1) zero mean

$$\mathrm{E}(\varepsilon_i) = 0,\ i = 1,2,\cdots,n \tag{53}$$

(2) constant variance (homoscedasticity)

$$V(\varepsilon_i) = \sigma^2,\ i = 1,2,\cdots,n \tag{54}$$

(3) mutually uncorrelated

$$\mathrm{E}\left[\varepsilon_i \varepsilon_j\right] = 0,\ i \neq j,\ i,j = 1,2,\cdots,n \tag{55}$$

**Assumption 3:** $x_1, x_2, \cdots, x_p$ are not random variables. They are fixed values of explanatory variables, with

$$\mathrm{E}(y \mid \underline{x}) = \underline{x}^t \underline{\beta}. \tag{56}$$

**Assumption 4:** $\varepsilon_i \overset{d}{\sim} N(0, \sigma^2),\ i = 1,2,\cdots,n$.

In matrix presentation, Eq. (52) can be written as

$$\underline{y} = X \underline{\beta} + \underline{\varepsilon} \tag{57}$$

where the design or regressor matrix $X$ is given by

$$X_{n\times(p+1)} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix} = \begin{bmatrix} \underline{1} & \underline{x}_1 & \cdots & \underline{x}_p \end{bmatrix}$$
$$\text{and}\quad \underline{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix},\quad \underline{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix},\ \underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}. \tag{58}$$

It is obvious that in the classical regression model the variance-covariance is $\sigma^2 I_{n\times n}$. The ordinary least square method will give the model estimation and analysis. The Gauss-Markov theorem guarantees the BLUE property (Myers, 2000; Rao, 1973) of the coefficient estimators. If the regression model variance-covariance is not $\sigma^2 I_{n\times n}$ but $\sigma^2 V_{n\times n}$, then the weighted least square estimation is necessary.

The basic uncertain canonical process regression model will take a conditional form

$$\begin{aligned} & y_i = \beta_0 + \beta_2 x_{1i} + \cdots + \beta_m x_{mi} + \sigma C_i \\ & i = 1, 2, \cdots, n \end{aligned} \tag{59}$$

where $y$ is the response variable, $x_1, x_2, \cdots, x_m$ are $m$ explanatory variables without uncertain influences, and $\sigma C_i$, $i = 1, 2, \cdots, n$ are the uncertain error terms from an uncertain canonical process $\{C_t, t \geq 0\}$.

The matrix version of Eq. (59) is

$$\underline{y} = X \underline{\beta} + \underline{C} \tag{60}$$

where

$$\begin{aligned} & \underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \; X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{m1} \\ 1 & x_{12} & \cdots & x_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1n} & \cdots & x_{mn} \end{bmatrix}, \\ & \underline{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}, \; \underline{C} = \begin{bmatrix} \sigma C_1 \\ \sigma C_2 \\ \vdots \\ \sigma C_n \end{bmatrix} \end{aligned} \tag{61}$$

**Theorem 32:** The basic uncertain regression model in Eq. (59) or Eq. (60) have the following properties:

(1) $\mathrm{E}[\underline{C}] = \underline{0}$;

(2) $\mathrm{E}\lfloor \underline{y} \rfloor = X \underline{\beta}$ ;

(3) $\Sigma = \mathrm{E}\lfloor \underline{C}\underline{C}^t \rfloor = \lfloor \sigma_{i,j} \rfloor_{n \times n}$, where $\sigma_{i,j} = i^2 + \delta_{i,j-i}$.

and thus the uncertain canonical process regression is a weighted regression model.

**Proof:** Based on the definition of an uncertain canonical process $\{C_t, t \geq 0\}$, $C_1, C_2, \cdots, C_n$ are members of the uncertain canonical process family, whose indices are integers $i = 1, 2, \cdots, n$, and therefore $C_i = C_i - C_0$ are uncertainty variables with an uncertainty distribution

$$\Psi_{C_i}(z) = \left(1 + \exp\left(-\frac{\pi z}{\sqrt{3} i}\right)\right)^{-1} \tag{62}$$

Then, the expectation $\mathrm{E}[C_i] = 0$ and the variance $V[C_i] = i^2$. Hence $\mathrm{E}[\underline{C}] = \underline{0}$ then $\mathrm{E}\lfloor \underline{y} \rfloor = X \underline{\beta}$ is proved.

As to Property (3), we noted the error vector is composed of $\sigma C_1, \sigma C_2, \cdots, \sigma C_n$ sequentially, it is simple to establish that

$$\Sigma = \mathrm{E}\left[\underline{C}\underline{C}^{'}\right] = \sigma^2 \left(\sigma_{ij}\right)_{n\times n} = \sigma^2 \left(ij + \delta_{i,j-i}\right)_{n\times n} \tag{63}$$

Thus, the uncertain canonical process regression model is weighted one since the variance-covariance is not $\sigma^2 I_{n\times n}$ but is $\sigma^2 \left(ij + \delta_{i,j-i}\right)_{n\times n}$.

**Remark 33:** From the model formulation and Theorem 32 statement and proof, it should be emphasized that a key feature of this basic uncertain regression model lies in the error term assumption: errors $\sigma C_1, \sigma C_2, \cdots, \sigma C_n$ are members of an uncertain canonical process, whose indices are $i = 1, 2, \cdots, n$. In classical regression model, the error terms are not taken sequentially. In other words, sequential order of error terms plays a fundamental role in uncertain regression. In that sense, it is logical to describe an uncertain regression model as a sequential regression model.

**Theorem 34:** (Gauss-Markov Theorem) The estimator of the coefficient vector in the basic uncertain regression defined by $\underline{\hat{\beta}} = \left(X^{'}\Sigma^{-1}X\right)^{-1} X^{'}\Sigma^{-1}\underline{y}$) is BLUE (Best - minimum variance, Linear, Unbiased Estimator).

**Proof:** The estimator of $\underline{\beta}$, $\underline{\hat{\beta}} = \left(X^{'}\Sigma^{-1}X\right)^{-1} X^{'}\Sigma^{-1}\underline{y}$, is linear in response values $y_2, y_2, \cdots, y_n$ as is obvious from the a $(m+1)\times n$ matrix pre-multiplier. The unbiasedness follows from:

$$\begin{aligned}
&\mathrm{E}\left[\underline{\hat{\beta}}\right] = \mathrm{E}\left[\left(X^{t}\Sigma^{-1}X\right)^{-1} X^{t}\Sigma^{-1}\underline{y}\right] \\
&= \left(X^{t}\Sigma^{-1}X\right)^{-1} X^{t}\Sigma^{-1}\mathrm{E}\left[\underline{y}\right] = \left(X^{t}\Sigma^{-1}X\right)^{-1} X^{t}\Sigma^{-1}X\underline{\beta} = \underline{\beta}
\end{aligned} \tag{64}$$

Then, let us assume another unbiased estimator $\underline{\tilde{\beta}} = A\underline{y} = \left(\left(X^{'}\Sigma^{-1}X\right)^{-1} X^{'}\Sigma^{-1} + D\right)$ then

$$\begin{aligned}
&\mathrm{E}\left[\underline{\tilde{\beta}}\right] \\
&= \mathrm{E}\left[\left(\left(X^{t}\Sigma^{-1}X\right)^{-1} X^{t}\Sigma^{-1} + D\right)\underline{y}\right] \\
&= \left(\left(X^{t}\Sigma^{-1}X\right)^{-1} X^{t}\Sigma^{-1} + D\right)\mathrm{E}\left[\underline{y}\right] \\
&= \left(\left(X^{t}\Sigma^{-1}X\right)^{-1} X^{t}\Sigma^{-1}X + DX\right)\underline{\beta} \\
&= \left(I + DX\right)\underline{\beta}
\end{aligned} \tag{65}$$

Then $DX = 0$ is implied. Now, let us examine the variance of $\underline{\tilde{\beta}}$,

$$V\left[\underline{\tilde{\beta}}\right]=V\left[\left(X^{t}\Sigma^{-1}X\right)^{-1}X^{t}\Sigma^{-1}\underline{y}+D\underline{y}\right]=V\left[\underline{\hat{\beta}}+D\underline{y}\right]$$
$$=V\left[\underline{\hat{\beta}}\right]+D\mathrm{E}\left[\underline{C}\underline{C}^{t}\right]D^{'}=V\left[\underline{\hat{\beta}}\right]+D\Sigma^{-1}D^{t}\geq V\left[\underline{\hat{\beta}}\right] \tag{66}$$

## 2.6 Gaussian process regression

Bayesian modeling (Chen, 1981; Lee, 1989; DeGroot, 1970; Ferguson, 1967; Guo, 2011) can avoid the large sample requirement by introducing a prior for the unknown parameters. Gaussian process regression (Williams and Rasmussen, 1996; Snelson et al., 2004; Rasmussen and Williams, 2006), is a Bayesian regression model which take the intrinsic variance-covariance of a Gaussian process as the prior. It presumes a covariance function

$$v\left(x,x'\right)=\sigma_f^2\exp\left(-\frac{\left|x-x'\right|^2}{2l^2}\right) \tag{67}$$

where the maximum allowable variance is $\sigma_f^2$. As $v(x,x')$ approaches $\sigma_f^2$, which implies $f(x)$ is perfectly associated with $f(x')$. Multivariate versions of Eq. (67) can be constructed using quadratic forms such as

$$v\left(\underline{x},\underline{x}'\right)=\sigma_f^2\exp\left(-\frac{\left(\underline{x}-\underline{x}'\right)^t AA^t\left(\underline{x}-\underline{x}'\right)}{2l^2}\right). \tag{68}$$

For the novel Gaussian process regression,

$$v\left(x,x'\right)=\sigma_f^2\exp\left(-\frac{\left|x-x'\right|^2}{2l^2}\right)+\sigma_n^2\delta\left(x,x'\right) \tag{69}$$

where $\delta(x,x')$ is the Kronecker delta function.

The Gaussian process regression takes a form

$$y=f\left(\underline{x}\right)+G_t \tag{70}$$

where $\{G_t, t\geq 0\}$ is a Gaussian process. With sample observations $\underline{y}$, and single observation $y_*$ to be predicted on the basis of the data and model, then we note

$$\begin{bmatrix}\underline{y}\\ y_*\end{bmatrix}\overset{d}{\sim}N\left(\begin{bmatrix}\underline{\mu}\\ \mu_*\end{bmatrix},\begin{bmatrix}V & V_*^t\\ V_* & V_{**}\end{bmatrix}\right) \tag{71}$$

where

$$V=\begin{bmatrix} v\left(x_1,x_1\right) & v\left(x_1,x_2\right) & \cdots & v\left(x_1,x_n\right) \\ v\left(x_2,x_1\right) & v\left(x_2,x_1\right) & \cdots & v\left(x_2,x_n\right) \\ \vdots & \vdots & \ddots & \vdots \\ v\left(x_n,x_1\right) & v\left(x_2,x_n\right) & \cdots & v\left(x_n,x_n\right) \end{bmatrix},$$
$$V_*=\left[v\left(x_*,x_1\right) \quad v\left(x_*,x_2\right) \quad \cdots \quad v\left(x_*,x_n\right)\right],$$
$$V_{**}=\left[v\left(x_*,x_*\right)\right]. \tag{72}$$

The prediction $y_* \mid \underline{y}$ satisfies

$$y_* \mid \underline{y} \overset{d}{\sim} N\left(\mu_* + V_* V^{-1}\left(\underline{y}-\underline{\mu}\right), V_{**} - V_* V^{-1} V_*^t\right). \tag{73}$$

If we wish to predict a vector $\underline{y}_* \mid \underline{y}$ whether without or with the presence in the model of a further p explanatory variables for the vector of means, along with any input variables determining known variance-covariance matrix V, then

$$\underline{y}_2 \mid \underline{y}_1 \overset{d}{\sim} N\left(X_2\underline{\beta} + V_{21}V_{11}^{-1}\left(\underline{y}_1 - X_1\underline{\beta}\right), V_{22} - V_{21}V_{11}^{-1}V_{12}\right). \tag{74}$$

With the knowledge of regression modeling assumptions, particularly, the Gaussian process regression model assumptions, we are ready to construct the next layer of preparation for the uncertain canonical process regression modeling - the variance-covariance structure of the multivariate uncertainty distribution for the uncertainty response observations $\underline{\xi}$.

## 2.7 Maximum likelihood estimation

Maximum likelihood is a typical approach for estimating the unknown parameters of a distributional family (Lawless, 1982; Blische and Murthy, 2000; Ushakov, 1994).

Let $L(\underline{\theta} \mid \mathrm{K}) = f\left(x_1, \vartheta_1, x_2, \vartheta_2, \cdots, x_N, \vartheta_N; \underline{\theta}\right)$ with $f\left(\cdot;\underline{\theta}\right)$ representing the joint distribution of data $\mathrm{K}$. This is then called the likelihood function with respect to parameter set $\underline{\theta}$, $\underline{\theta} \in \Theta$.

**Definition 35:** Let $\mathrm{K} = \left\{\left(x_i, \vartheta_i\right), i = 1, 2, \cdots, N\right\}$ be a failure-censoring data record, i.e.,

$$\vartheta_i = \begin{cases} 0 & x_i \text{ is a natural failure} \\ 1 & x_i \text{ is a censored event} \end{cases} \tag{75}$$

then

$$L\left(\underline{\theta} \mid \mathrm{K}\right) = \prod_{i=1}^{N} f^{1-\vartheta_i}\left(x_i;\underline{\theta}\right) R^{\vartheta_i}\left(x_i;\underline{\theta}\right) \tag{76}$$

where $f$ is the failure density function and $R$ is the reliability function.

**Definition 36:** The function

$$l(\underline{\theta}\,|\,\mathrm{K}) = \ln\left(L(\underline{\theta}\,|\,\mathrm{K})\right) \tag{77}$$

is called the log-likelihood function.

**Lemma 37:** $\underline{\theta}_0$ is an optimal point for $l(\underline{\theta}\,|\,\mathrm{K})$ if and only if it is an optimal point for $L(\underline{\theta}\,|\,\mathrm{K})$.

Note that $\ln(\cdot)$, whose base is $e > 1$, is monotone increasing. Therefore the patterns in $L(\underline{\theta}\,|\,\mathrm{K})$ will be well-maintained by $l(\underline{\theta}\,|\,\mathrm{K})$ and the converse is also true:

$$l(\underline{\theta}_0\,|\,\mathrm{K}) = \max\left\{l(\underline{\theta}\,|\,\mathrm{K})\right\} \Leftrightarrow L(\underline{\theta}_0\,|\,\mathrm{K}) = \max\left\{L(\underline{\theta}\,|\,\mathrm{K})\right\} \tag{78}$$

Turning our attention now to wave-like lifetime distribution of Type I, it has a form:

$$F(x) = 1 - \exp\left(-\int_0^x \left(\gamma + \frac{\sin^2 \alpha s}{s^2}\right) ds\right) \tag{79}$$

with two-parameter hazard function

$$h(x) = \gamma + \frac{\sin^2 \alpha x}{x^2},\ x \in [0, +\infty),\ \alpha > 0,\ \gamma \geq 0 \tag{80}$$

**Theorem 38:** For the Type I wave-like distribution, the log-likelihood function is

$$l(\alpha, \gamma\,|\,\mathrm{K}) = \sum_{i=1}^{N} (1 - \vartheta_i) \ln\left(\gamma + \frac{\sin^2 \alpha x_i}{x_i^2}\right) - \sum_{i=1}^{N} \int_0^{x_i} \left(\gamma + \frac{\sin^2 \alpha s}{s^2}\right) ds \tag{81}$$

The first-order partial derivatives are

$$\begin{aligned}
\frac{\partial l(\alpha,\gamma\,|\,\mathrm{K})}{\partial \alpha} &= \sum_{i=1}^{N} \frac{\sin(2\alpha x_i)(1-\vartheta_i) x_i}{\gamma x_i^2 + \sin^2 \alpha x_i} - \sum_{i=1}^{N} \int_0^{x_i} \frac{\sin(2\alpha s)}{s} ds \\
\frac{\partial l(\alpha,\gamma\,|\,\mathrm{K})}{\partial \gamma} &= \sum_{i=1}^{N} \frac{x_i^2 (1-\vartheta_i)}{\gamma x_i^2 + \sin^2 \alpha x_i} - \sum_{i=1}^{N} x_i
\end{aligned} \tag{82}$$

And the second-order order partial derivatives are

$$\begin{aligned}
\frac{\partial^2 l(\alpha,\gamma\,|\,\mathrm{K})}{\partial^2 \alpha} &= \sum_{i=1}^{N} \frac{2\cos(2\alpha x_i)\left(\gamma x_i^2 + \sin^2 \alpha x_i\right) - \sin^2(2\alpha x_i)}{\left(\gamma x_i^2 + \sin^2 \alpha x_i\right)^2} (1-\vartheta_i) x_i^2 \\
&\quad - \frac{1}{\alpha} \sum_{i=1}^{N} \sin(2\alpha x_i) \\
\frac{\partial^2 l(\alpha,\gamma\,|\,\mathrm{K})}{\partial \alpha \partial \gamma} &= -\sum_{i=1}^{N} \frac{x_i^3 (1-\vartheta_i) \sin(2\alpha x_i)}{\left(\gamma x_i^2 + \sin^2 \alpha x_i\right)^2} \\
\frac{\partial^2 l(\alpha,\gamma\,|\,\mathrm{K})}{\partial \gamma^2} &= -\sum_{i=1}^{N} \frac{x_i^4 (1-\vartheta_i)}{\left(\gamma x_i^2 + \sin^2 \alpha x_i\right)^2}
\end{aligned} \tag{83}$$

**Theorem 39:** For the Type II wave-like lifetime distribution with 2 parameters, and a hazard function of the form $h(x) = \gamma + \sin(\alpha x)/x$, the log-likelihood function in the presence of both failures and censored data is

$$l(\alpha,\gamma \mid \mathrm{K}) = \sum_{i=1}^{N}(1-\vartheta_i)\ln\left(\gamma + \frac{\sin(\alpha x_i)}{x_i}\right) - \gamma\sum_{i=1}^{N} x_i - \sum_{i=1}^{N}\int_0^{x_i}\frac{\sin(\alpha s)}{s}ds \tag{84}$$

The first-order partial derivatives are

$$\begin{aligned}
\frac{\partial l(\alpha,\gamma \mid \mathrm{K})}{\partial\alpha} &= \sum_{i=1}^{N}(1-\vartheta_i)\frac{x_i\cos(\alpha x_i)}{\gamma x_i + \sin(\alpha x_i)} + \frac{1}{\alpha}\sum_{i=1}^{N}\sin(\alpha x_i) \\
\frac{\partial l(\alpha,\gamma \mid \mathrm{K})}{\partial\gamma} &= \sum_{i=1}^{N}(1-\vartheta_i)\frac{x_i}{\gamma x_i + \sin(\alpha x_i)} - \sum_{i=1}^{N} x_i
\end{aligned} \tag{85}$$

And the second-order partial derivatives are

$$\begin{aligned}
\frac{\partial^2 l(\alpha,\gamma \mid \mathrm{K})}{\partial\alpha^2} &= -\sum_{i=1}^{N}(1-\vartheta_i)x_i^2\frac{\gamma x_i\sin(\alpha x_i)+1}{(\gamma x_i + \sin(\alpha x_i))^2} \\
&\quad - \frac{1}{\alpha^2}\sum_{i=1}^{N}\sin(\alpha x_i) + \frac{1}{\alpha}\sum_{i=1}^{N} x_i\cos(\alpha x_i) \\
\frac{\partial^2 l(\alpha,\gamma \mid \mathrm{K})}{\partial\alpha\partial\gamma} &= -\sum_{i=1}^{N}\frac{(1-\vartheta_i)x_i^2\cos(\alpha x_i)}{(\gamma x_i + \sin(\alpha x_i))^2} \\
\frac{\partial^2 l(\alpha,\gamma \mid \mathrm{K})}{\partial\gamma^2} &= -\sum_{i=1}^{N}\frac{(1-\vartheta_i)x_i^2}{(\gamma x_i + \sin(\alpha x_i))^2}
\end{aligned} \tag{86}$$

For more detailed developments of maximum likelihood estimation, see Lawless (1982), and Guo et al. (2010).

## 2.8 Genetic Algorithm

Genetic algorithm is the most frequently used optimization scheme in our early stage's investigations on the grey differential equation modeling practices. Meta-heuristic scheme is a class of optimization computational approaches, which engage searching very large spaces of candidate solutions, by iteratively improve a candidate solution or a group of candidate solutions quality, step by step to reach the final optimal solution, although a meta-heuristic scheme cannot guarantee the optimal solution to be global one (Fraser and Donald, 1970; Weise, 2008).

Standing on string related optimization scheme view of point, GA uses two-element membership set $\Theta = \{0,1\}$ and thus a binary number system is utilized to represent real numbers in $\mathbb{R} = (-\infty,+\infty)$. The binary strings are also called chromosomes, the collection of the binary strings is called the population of genetic algorithms. The GA evaluates each chromosome's fitness values at each generation (loop) of current population according to the objective function $f\left(\underline{x}\right)$, for example. Then GA selects chromosomes according their fitness values to breed a new generation. In reproduction procedure, GA takes crossover, mutation operations to make the new individuals. The six important concepts used and the roles played in a simple generational GA scheme are briefly listed and discussed as following:

**1. Initialization.** Given an objective function $f\left(\underline{x}\right)$, how many bits standard as a variable are necessary to decided: where the first bit of each unit stands as sign (positive or negative). If we choose $l$ bits representing each variable, for an $n$ dimensional objective function, the length of each chromosome is $length = l * n$. According to the nature of the optimization problem we investigated, the initial population size needs to specify, typically the initial population may contain hundreds or thousands chromosomes, from which, each chromosome of the population can be randomly selected.

**2. Evaluation.** GA evaluates chromosome's fitness values according to the objective function $f\left(\underline{x}\right)$ which already given.

**3. Selection.** According to the fitness values, GA selects chromosomes in the population for entering reproduction stage.

**4. Crossover.** Given two individual chromosomes, which are named as "parents", then, we randomly choose a location in both chromosomes, after that location all the remaining locations exchange between corresponding location s in the two selected two parents. For example, parent one chromosome is 11001111, parent two chromosome is 00110000. If we choose 4$^{th}$ bit as the location, the new produced two offspring become as 11000000, 00111111, after crossover.

**5. Mutation.** This operation of GA is defined as randomly flips some of the bits in a chromosome. For example, if the chromosome is 11001111, and the mutation operation occurs in second location of it, then, the mutated chromosome becomes 10001111.

**6. Reproductions.** This procedure is used to generate the next generation of population chromosomes, in which many selected paired parents', chromosomes according to "crossover" and "mutation" to generate their "babies". The baby's chromosomes have some characteristics from their "parents", but the fitness values improved or not which depend on the next loop's evaluation test. According to existing GA literature, three or more parents are more suitable for reproducing the next generation of chromosomes.

Finally, we state the flow chart of a simple GA scheme:

1. Initialization of GA. In this step, it decides how many bits are used to represent each variable, and initial whole the population of chromosomes randomly;

2. Start loop;

3 According the objective function, the evaluation of each chromosome's fitness values of whole population will perform. At the end of the evaluation, determine the chromosome achieved or not sufficient fitness values. If achieved, quite the loop (i.e., not continued);

4. Select more fit chromosomes as "parents" for reproduction;

5. Reproduce new "babies" chromosomes according to crossover and mutation operation to their selected "parents";

6. If the fitness value criterion is reached, stop, otherwise, return to step 2 to continue.

# Chapter 3. Collection of Papers

This chapter consists of consists of seven peer-reviewed journal papers, and two of them are EI indexed. Four journal papers contribute to development of the new global optimization algorithm-lambda algorithm, and two journal papers builds up the uncertainty decision theory and modelling to deal with general uncertainty data, and one journal paper uses the new computational methods to develop differential equation associated regression (DEAR) model.

List of peer-reviewed journal publications:

1. Cui, Y. H., Guo, R., and Guo, D. (2009). A Naive Five-Element String Algorithm. Vol. 4, No. 9, pp 925-934, November 2009. Journal of Software, ISSN 1796-217X, Academic Publisher. doi: 10.4304/jsw.4.9.925-934. (*published*)
2. Cui, Y.H., Guo, R., and Guo, D. (2010). Lambda Algorithm. Vol. 4, No. 1, pp. 22-33. 2010. Journal of Uncertain Systems, ISSN: 1752-8909 (PRINT), ISSN: 1752-8917 (ONLINE), Publisher: World Academic Union. (*published*)
3. Cui, Y.H., Guo, R. (2011). Nash-lambda Algorithm with Application in Safety and Reliability. Vol. 1, pp. 51-58. 2011. Summer Safety and Reliability Seminars, SSARS 2011, Journal of Polish Safety and Reliability Association, ISBN: 978-83-925436-2-6. Editors: Krzysztof Kolowrocki, Joanna Soszynska and Enrico Zio. (*published*)
4. Cui, Y.H., Guo, R. (2011). Lambda Algorithm and Maximum likelihood Estimation. Vol. 1, pp. 59-72. 2011. Summer Safety and Reliability Seminars, SSARS 2011, Journal of Polish Safety and Reliability Association, ISBN: 978-83-925436-2-6. Editors in Chief: Krzysztof Kolowrocki, Joanna Soszynska and Enrico Zio. (*published*)
5. Cui, Y.H., Guo, R., Dunne, T., and Guo, D. (2010). Decision Theory under General Uncertainty. Vol. 1, pp 51 – 66. 2010. Summer Safety and Reliability Seminars, SSARS 2010, Journal of Polish Safety and Reliability Association, ISBN: 978-83-925436-2-6. Editors: Krzysztof Kolowrocki, Joanna Soszynska and Enrico Zio. (*published*)
6. Cui, Y.H. Guo, R., Dunne, T., and Guo, D. (2011). Bayesian Uncertainty Decision Analysis. Vol. 2, No. 1 (20), pp 70-81, March 2011. Reliability & Risk Analysis: Theory & Applications. (Electronic Journal of International Group on Reliability – Gnedenko E-Forum). (ISSN: 1932-2321) Website: http://www.gnedenko-forum.org/Journal/ (*published*)

7. Cui, Y.H., Guo, R., and Guo, D. (2011). Probabilistic DEAR Modeling. International Journal of Machine Learning and Cybernetics. (*under review*)

## 3.1 A Naïve Five-Element String Algorithm

Yanhong Cui, Renkuan Guo

Department of Statistical Sciences, University of Cape Town

Private Bag, Rhodes' Gift, Rondebosch 7701, Cape Town, South Africa

hadisna@hotmail.com renkuan.guo@uct.ac.za

Danni Guo

South African National Biodiversity Institute,

Private Bag X7, Claremont 7735, Cape Town, South Africa

d.guo@sanbi.org.za



This paper has been published in the Journal of Software, Vol. 4, No. 9, pp 925-934, November 2009. ISSN 1796-217X, Academic Publisher. doi: 10.4304/jsw.4.9.925-934.

### 3.1.1 Abstract

In this paper, we propose a new global optimization algorithm inspired by the human life model in Chinese Traditional Medicine and graph theory, which is named as naïve five-element string algorithm. The new algorithm utilizes strings of elements from member set {0, 1, 2, 3, and 4} to represent the values of candidate solutions (typically represented as vectors in n-dimensional Euclidean space). Except the mathematical operations for evaluating the objective function, sort procedure, creating initial population randomly, the algorithm only involves if-else logical operation. In contrast to existing global optimization algorithms, the five-element algorithm engages the simplest mathematics but reaches the highest searching efficiency.

**Keywords:** global optimization, five-element string, genetic algorithm, sort, naïve string algorithm



### 3.1.2 Introduction

Optimization is one of challenging and active mathematical research branches. Particularly, the topic of global optimization is of critical importance because of the high demand and wide applications in science, business, economy, and industry.

Natural world is always the best teacher since the human society started, for example, fire usage, planting crops, hunting animals, Kungfu exercises, and so on. Many global optimization algorithms are imitating the behavior of biological world, for example, genetic algorithm (abbreviated as GA), ant colony algorithm, monkey algorithm, etc. However, those biological imitated algorithms are direct “copy” of natural evolution. It is necessary to mention that ancient scientists and philosophers had established many abstract biological models for human life and natural living beings, for example, Yin-Yang and five elements (Wu-Xing) are among them. Yin-Yang and five elements (Wu-Xing) doctrines are still active in guiding today’s Traditional Chinese Medicine (abbreviated as TCM practices).

In today’s part-theory dominated western scientific communities, exploration of an ancient Chinese philosophy and its potential scientific value is often regarded as nonsense, pseudo-science or waste of times. Scientists more tend to believe part-theory based genetic engineering rather than TCM. We are not resisting any advancement in science and technology, however, we also have to accept the cruel realities: only 30% of the patients or illness could be cured by modern western medicine, and on other hand, no less than 30% of the patients or illness could be cured by traditional medical treatments. More and more people accept traditional medical treatments because of cost-saving and effectiveness, for example, acupuncture and moxibustion from TCM. The theoretical foundation guiding TCM is ancient Chinese Yin-yang and Wu-Xing doctrines. Yinyang concept plays roles in other scientific fields too. The link between Yin-Yang representation and binary number system is already well-known, and it is not difficult to reveal certain root of GA in Yin-Yang doctrine. We notice that a common rule guiding TCM doctors: identifying and eliminating factors causing in-balances within patient’s body system and strengthening these factors leading the patient to his/her harmonious state according to five-element (Wu-Xing) doctrine. A natural question arises inevitably: is it possible to create an algorithm for searching global optimum with a root in five-element (Wu-Xing) doctrine? A faith inspires us is that TCM medical exercises are nothing but seek human body system optimal state, with the aid of Yin-Yang, five-element (Wu-Xing) system model.

The five-element doctrine (Wu-Xing) is different from Greek theory of four elements in formality, but both are atomic theory of substances. Ancient philosophers believed that five elements: „metal’, ‘wood’, ‘water’, „fire’, „earth’, constitute of the world. People started using the features of five-element to explain the changing of the object world in terms of the five-element’s generation and deduction relationship for evolving into next sub-balanced state. However, the ever-changing nature of object world would repeatedly evolutions along the direction of generation and deduction until the system reaches its intrinsic harmonious state, even it is temporary but relatively stable. In other words, theory of five elements (Wu-Xing) as a Chinese ancient philosophy was not a merely five-substance constitution of existing objects or systems surround us but more critically the theory of five elements provides the guidance for people to seek the intrinsic harmonious state of a system under investigation. This is the reason why Traditional Chinese Medicine is using the so-called Five-Element Doctrine as its foundation because an individual human being in good health is nothing but is in a harmonious state.

Definitely, our new five-element global optimization searching algorithm is not simulating ancient five element objects, instead, we has established a dedicated link between mathematical objective function under investigation and the simulated population of five element strings help the accomplishment of the optimal solution searching. In other words, the five-element algorithm followed the idea of computer simulation, not only a simple mimicking of some natural phenomenon, but also a creative idea generates from old Chinese traditional Wu-Xing Doctrine.

The five-element algorithm treat the object function as a system, by simulating the five-element strings involved in this system followed by cycles of balance to generates a better system or find a better solution of object function.

It is necessary to mention here that many existing global optimization algorithms engage complicated mathematical operations, for example, algebraic operators, derivative operator, integration operator, projection operator (for parameter calibration) and control operator, however, five-element algorithm only engages the simplest logical operator: if-else. This feature greatly saves the computing time. The adjective “naïve” is added for reminding that this new algorithm does not involve complicated.

### 3.1.3 An Inspiring Example

The objective function for illustrating purpose is the Rosenbrok function

$$f(x,y)=100\left(y-x^2\right)^2+\left(1-x\right)^2 \tag{1}$$

The global minimal value is 0 of Rosenbrok function at $(x,y)=(1,1)$. The plot of (1) in Figure 3.1.1 offers an intuitive view on the features optimality of (1).

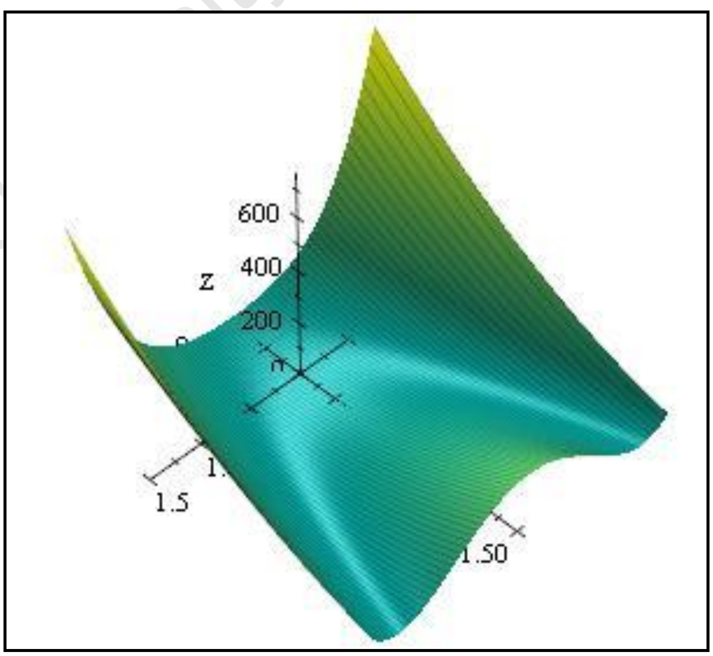



Figure 3.1.1 Plot of Rosenbrok function

We used GA (Matlab help Example, 2008) to search the global minimum of Rosenbrok function. However, contrary to the global optimization searched by GA, the "global" minimal value 0.11935 at $(x,y)=(0.681,0.477)$ is reported.

The case of GA's "failure" to search the true global minimum (for given computing time) here inspires us to consider the fundamental weakness of GA. It is noticed that GA, as a global optimization algorithm, differs from many other algorithms.

Let $f(\underline{x})$ be the objective function, where $\underline{x}=(x_1,x_2)^T \in \mathbb{D} \subset \mathbb{R}^2$. In many optimization searching algorithms, a typical exercise is trying to improve the optimality within the neighbourhood. It is obvious that the increment in $\underline{x}$ approach typically leads to a local optimum.

GA does not work on system state $\underline{x} \in \mathbb{D} \subset \mathbb{R}^n$ of the objective function $f(\underline{x})$ directly, rather it uses string like $0011001100100111010111100$ for the representing the state $\underline{x}^T = (x_1, x_2, \cdots, x_n)$ and hence may possess better global coverage. However, GA string member set is $\{0,1\}$. Inevitably, the change in string may not change the state $\underline{x}^T = (x_1, x_2, \cdots, x_n)$ efficiently for covering the whole domain because the element change in a string is 1.



### 3.1.4 String Representation of The State of Objective Function

An improvement strategy is to expand string member set. Now let us formally establish the string representation related concepts.

**Definition 1:** A string is a sequence of integers, denoted by $n_1 n_2 \cdots n_p$. Number $p$ is called the length of a string.

For operational convenience, a string may be repressed by a row vector, $\underline{n}^T = (n_1, n_2, \cdots, n_p)$.

**Definition 2:** The collection of the elements for construction a string, denoted by $\{0,1,2,\cdots,s-1\}$, is termed as an element set for a string. $s$ is called the size of the element set of the string (i.e., the number of elements in element set).

**Conjecture 3:** The size of the element set of a string used in naïve string algorithm is a prime number.

In GA, the size of the element set $\{0,1\}$ is prime number 2. Prime number 3, 5, 7, 11, etc can also be used. If the size of the element set is 7, then the element set is $\{0,1,2,3,4,5,6\}$. The length of a string $p$ should be at least $n(s+1)$.

**Definition 4:** Let $\underline{x} = (x_1, x_2, \cdots, x_n)^T \in \mathbb{D} \subset \mathbb{R}^2$, denote system state, which is also representing the candidate solution. Then the length of the string representing $\underline{x}$ is $p > nu$ if the size of the string element set is $s$, $u > s$, $u$ is called the basic unit size of a string. The string representation for $(x_1, x_2, \cdots, x_n)^T$ is

$$e_1 e_2 \cdots e_u e_{u+1} \cdots e_{2u} \cdots e_{(n-1)u+1} \cdots e_{nu} \tag{2}$$

An intuitive correspondence between the state $\underline{x}$ and the representing string is

$$\underbrace{e_1 e_2 \cdots e_u}_{x_1} \underbrace{e_{u+1} \cdots e_{2u}}_{x_2} \cdots \underbrace{e_{(n-1)u+1} \cdots e_{nu}}_{x_n} \tag{3}$$

**Lemma 5:** Let the system state be $\underline{x} \in \mathbb{D} \subset \mathbb{R}^n$, and $\underline{e}$ be a string representation (of the system state) with element set size $s$ and string length $n(s+1)$. Let $\underline{u}_{\min} \le \underline{x} \in \mathbb{D}$, $\underline{u}_{\max} \ge \underline{x} \in \mathbb{D}$, and $u_r = \max_{1 \le i \le n}(u_{\max,i} - u_{\min,i})$. The weight matrix $O = (o_{ij})_{n \times nu} = (\underline{o}_1^T, \underline{o}_2^T, \cdots, \underline{o}_n^T)^T$ with the $i^{th}$ row vector $\underline{o}_i^T$ having a form

$$\left(0, 0, \cdots, 0, \cdots, \frac{s^s}{s^{s+1}}, \frac{s^{s-1}}{s^{s+1}}, \cdots, \frac{s^0}{s^{s+1}}, \cdots, 0, 0, \cdots, 0\right) \tag{4}$$

where the nonzero weights are located at the $i^{th}$ segment. Then the system state is a linear transformation of the $s$-element string representation

$$\underline{x} = \underline{u}_{\min} + u_r O \underline{e} \tag{5}$$

**Definition 6:** If $e$ is an element of a string with element set $\{0,1,2,3,\cdots,s-1\}$, then the value changing rule is

$$e = \begin{cases} e+1 & \text{if } e \in \{0,1,2,\cdots s-2\} \\ 0 & \text{if } e = s-1 \end{cases} \tag{6}$$

In the remaining sections of this paper, we will use 5-element string for illustration and the establishment of the naïve string algorithm.

### 3.1.5 Five-element String Representation

For clarity, we will use numerical examples for illustrating the necessity and advantages of string representation.

**Example 7:** Let $\mathbb{D} \triangleq u_{\min}, u_{\max} \times u_{\min}, u_{\max}$ be the domain for an objective function $f\ x_1, x_2$ . Assume that a string 1 2 4 3 0 1 2 4 3 2 11 represents $x_1, x_2$ : the first 6 elements, i.e., 1 2 4 3 0 1, in the string stand as $x_1$ and the second 6 elements, i.e., 2 4 3 2 11, stand as $x_2$. The element set is $0,1,2,3,4$ , the size of element set is $5$, the basic unit $u = 5 + 1 = 6$. The length of the string 1 2 4 3 0 1 2 4 3 2 11 is $2u$ =12, which is the number of units occupied in computer.

Mathematically, the linear system linking the five-element string and the system state can be expressed by

$$\begin{cases} x_1 = u_{\min} + u_{\max} - u_{\min} \sum_{j=1}^{6} e_j \dfrac{5^{6-j}}{5^6} \\ x_2 = u_{\min} + u_{\max} - u_{\min} \sum_{j=7}^{12} e_j \dfrac{5^{12-j}}{5^6} \end{cases} \tag{7}$$

Let

$$O_{2\times 2u} = \begin{bmatrix} \dfrac{5^5}{5^6} & \cdots & \dfrac{5^0}{5^6} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \dfrac{5^5}{5^s} & \cdots & \dfrac{5^0}{5^s} \end{bmatrix} \quad \underline{e}_{2u\times 1} = \begin{bmatrix} e_1 \\ \vdots \\ e_6 \\ e_7 \\ \vdots \\ e_{12} \end{bmatrix} \tag{8}$$

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \underline{u}_{\min} = \begin{bmatrix} u_{\min} \\ u_{\min} \end{bmatrix} \quad u_r = u_{\max} - u_{\min}$$

Then a matrix equation for string to state vector transformation is

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} u_{\min} \\ u_{\min} \end{bmatrix} + (u_{\max} - u_{\min}) \begin{bmatrix} \dfrac{5^5}{5^6} & \cdots & \dfrac{5^0}{5^6} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \dfrac{5^5}{5^6} & \cdots & \dfrac{5^0}{5^6} \end{bmatrix} \begin{bmatrix} e_1 \\ \vdots \\ e_6 \\ e_7 \\ \vdots \\ e_{12} \end{bmatrix} \tag{9}$$

Matrix $O$ is actually a weighting system which promotes the changes in $x_1, x_2$ according to the location of an individual member in the string as well as the changing size of the member.

In other words, the mechanism underlying the usage of string lies on that the weighting system, i.e., $\left\{\frac{5^5}{5^6}, \frac{5^4}{5^6}, \frac{5^3}{5^6}, \frac{5^2}{5^6}, \frac{5^1}{5^6}, \frac{5^0}{5^6}, 0,0,0,0,0,0\right\}$, assigned to the 6 members in the first half of the string and $\left\{0,0,0,0,0,0, \frac{5^5}{5^6}, \frac{5^4}{5^6}, \frac{5^3}{5^6}, \frac{5^2}{5^6}, \frac{5^1}{5^6}, \frac{5^0}{5^6}\right\}$, the weighting system assigned to the 6 members in the second half of the string create the possibility that change in the member of the string will have different impacts.

A string, denoted by $e_1 e_2 \cdots e_6 e_7, e_8 \cdots e_{12}$, the blue-color members are the first half of the string, representing $x_1$, the red-color members are the second half of the string, representing $x_2$. Logically, changes in $e_1$ and $e_7$ will result in large changes in $x_1$ and $x_2$ respectively, because the highest weight 0.2 is assigned to them, while changes in $e_6$ and $e_{12}$ will result in the smallest changes in $x_1$ and $x_2$ respectively, because the lowest weight 0.000064 is assigned to them. Therefore, a well-constructed string element shift scheme will have a balanced global searching capability as well as local fine-tune capacity.

**Example 8:** (Continued) Define $P(E|H)$, $u_{\max} = +10^{10}$, then $u_r = u_{\max} - u_{\min} = 2\times10^{10}$. String 1: 1 2 4 3 0 1 2 4 3 2 11 used in Example 7 is the base for observing the impacts from string member changes. String 2 changes the first element of the String 1 by adding 1 and the seventh element of the String 1 by adding 1, which is the smallest shift in size at highest weight 0.2. The change in $x_1$ and $x_2$ is quite large with distance 5656854249.5. However, String 3 changes the sixth element of the String 1 by adding 3 and the seventh element of the String 1 by adding 3, which is the largest shift in size at highest weight 0.000064. The change in $x_1$ and $x_2$ is much small with distance 202276452.4. Table 3.1.1 summaries the changes and impacts.

Table 3.1.1 The Impacts of Weights in Global Searching and Local Tune-up

| String | $x_1$ | $x_2$ | $\lvert\Delta x\rvert$ |
|---|---|---|---|
| 1 2 4 3 0 1 2 4 3 2 1 1 | -3662720000 | 1751680000 | |
| 2 2 4 3 0 1 3 4 3 2 1 1 | 337280000 | 5751680000 | 5656854249.5 |
| 1 2 4 3 0 4 2 4 3 2 1 4 | -3460480000 | 1755520000 | 202276452.4 |

It is important to emphasize here that the value of a string depends on three factors: (1) value of individual element in a string from {0, 1, 2, 3, 4}; (2) the location (or position) of a specific element i e ; (3) the combination of all elements appeared in the given string. Formally, let us define the five-element if-else operator, called as λ operator.

**Definition 9:** ($\lambda$ operator) Let $e \in \{0,1,2,3,4\}$, then

$$\lambda[e] = \begin{cases} e+1 & \text{if } e \in \{0,1,2,3,4\} \\ 0 & \text{if } e = 4 \end{cases} \tag{10}$$

$\lambda^{(l)}[\ ]$ is $l^{th}$ order $\lambda$ operator, which repeats $\lambda$ operation $m$ times.

**Definition 10:** (Modulo operator) Let $d$ be a positive integer, $q$ be the quotient and $r$ remainder $r$ satisfying

$$d = nq + r \tag{11}$$

Then we write the modulo operation as

$$d \bmod(q) = r \tag{12}$$

**Definition 11:** Let $\underline{e} = (e_1, e_2, \cdots, e_g)$ be a five-element string, then the $\lambda$ operation on a string is a component-wise operation, i.e.

$$\lambda[\underline{e}] = (\lambda[e_1], \lambda[e_2], \cdots, \lambda[e_g]) \tag{13}$$

Furthermore, let $A = (e_{ij})_{h \times g}$ be a five-element matrix, i.e., $e_{ij} \in \{0,1,2,3,4\}$, then

$$\lambda[A] = (\lambda[e_{ij}])_{h \times g} \tag{14}$$

**Proposition 12:** $\lambda^{(l)}[e] = \lambda^{(l \bmod(4))}[e]$, where $\lambda^{(0)}[e] \triangleq e$.

**Proof:** Note that $e \in \{0,1,2,3,4\}$, the number $e$ only has five choices. For example, $e = 0$,

$$\lambda^{(5)}[e] = \lambda^{(4)}[1] = \lambda^{(3)}[2] = \lambda^{(2)}[3] = \lambda^{(1)}[4] = 0 \tag{15}$$

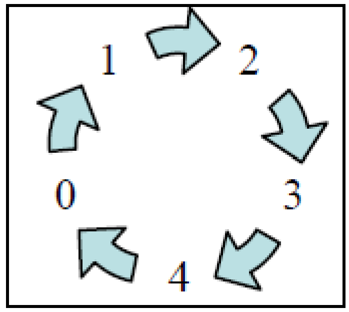



Figure 3.1.2 $\lambda[\ ]$ operation cycle

For any element $e \in \{0,1,2,3,4\}$, one-time $\lambda[\ ]$ operation shifts the element $e$ from current position into the next $1^{st}$ position along the cycle shown in Figure 3.1.2. Hence $l-$time $\lambda[\ ]$ operation shifts the element $e$ from current position into the next $l^{th}$-position along the cycle. Further, due to the fact that five-element member set $\{0,1,2,3,4\}$ only has five members in it, the period of the cycle is 5. Therefore, $\lambda^{(l)}[e] = \lambda^{(l\ \mathrm{mod}(4))}[e]$ since 0 is the first member of the element set.

**Proposition 13:** For any given five-element string $\underline{e}$, the five-time $\lambda[\ ]$ operated strings form a string cycle. In other words, $\left\{\underline{e}, \lambda^{(1)}[\underline{e}], \lambda^{(2)}[\underline{e}], \lambda^{(3)}[\underline{e}], \lambda^{(4)}[\underline{e}]\right\}$ is a string cycle.

**Definition 14:** Let

$$\begin{aligned} &\underline{x}^{(k)} = \underline{u}_{\min} + u_r O \lambda^{(k)}[\underline{e}], \\ &k = 0,1,2,3,4 \end{aligned} \tag{16}$$

Be the corresponding system state of $\lambda^{(k)}[\underline{e}]$. Then $\left\{\underline{x}, \underline{x}^{(1)}, \underline{x}^{(2)}, \underline{x}^{(3)}, \underline{x}^{(4)}\right\}$ is the system state cycle and $\left\{f(\underline{x}), f\left(\underline{x}^{(1)}\right), f\left(\underline{x}^{(2)}\right), f\left(\underline{x}^{(3)}\right), f\left(\underline{x}^{(4)}\right)\right\}$ is the objective function value cycle respect to the string cycle $\left\{\underline{e}, \lambda^{(1)}[\underline{e}], \lambda^{(2)}[\underline{e}], \lambda^{(3)}[\underline{e}], \lambda^{(4)}[\underline{e}]\right\}$.

**Proposition 15:** Let

$$\begin{aligned} f_{\min} &= \min\left\{f(\underline{x}), f\left(\underline{x}^{(1)}\right), f\left(\underline{x}^{(2)}\right), f\left(\underline{x}^{(3)}\right), f\left(\underline{x}^{(4)}\right)\right\}, \\ f_{\max} &= \max\left\{f(\underline{x}), f\left(\underline{x}^{(1)}\right), f\left(\underline{x}^{(2)}\right), f\left(\underline{x}^{(3)}\right), f\left(\underline{x}^{(4)}\right)\right\} \end{aligned} \tag{17}$$

Then the objective function cycle will demonstrate three patterns: (i) $f\left(\underline{x}\right)=f_{\min}$, i.e., the remaining four objective function values are above the cycle starting value $f\left(\underline{x}\right)$; (ii) $f\left(\underline{x}\right)=f_{\max}$, i.e., the cycle starting value $f\left(\underline{x}\right)$; (iii) $f_{\min}\leq f\left(\underline{x}\right)\leq f_{\max}$, i.e., the cycle starting value $f\left(\underline{x}\right)$ falls between cycle minimum and maximum.

**Proposition 16:** The weight matrix $O$ in string and system state linking equation $\underline{x}=\underline{u}_{\min}+u_r O\underline{e}$ reveals the ever-changing and controllable character of five element string representation. And the three cycle patterns of objective function values with respect to string cycles reveal that $\lambda[\;]$ operations guarantee the chance for global optimum searching.

Now it is ready to state the scheme of the naïve five-element string algorithm.

### 3.1.6 A Naïve Global Optimum Search Scheme

A few terms are defined first.

| | |
|---|---|
| Stopping time | The algorithm stops after running for an amount of time in seconds, which is specified as stopping time |
| Population size | The population size defines numbers of rows of matrices, denoted by $N$ |
| String length | The string length defines the number of elements in each five-element string |
| n | the lower bound value of input variable |
| $u_{\min}$ | the lower bound value of input variables |
| $u_{\max}$ | **:** the upper bound value of input variables |

Before the searching scheme enters algorithm loop the naïve nature of the scheme requires the creation of a candidate solution string population. Randomly select numbers from member set $\{0,1,2,3,4\}$ uniformly and independently and put them into strings until the string population is established. It is obvious that the discrete uniform random number nature eliminates any possible bias for the starting the algorithm.

**Stochastic initialization:** Randomly generate 2N**,** say, $N=100$, five-element strings as candidate solutions, then divide the candidate solutions into two string vectors (two matrices of elements), the first string vector is denoted by $Q_{\min}$ and the second by $Q_{\max}$. The searching range for the $i^{th}$ component of system state $\underline{x}$ is $[u_{\min}, u_{\max}]$, i.e., $u_{\min} \le x_i \le u_{\max}$.

**Searching loop:**

**Step 1: 2N string cycles creation.** By applying $\lambda[\ ]$ to $Q_{\min}$ and $Q_{\max}$ respectively, ten string vectors (including $Q_{\min}$ and $Q_{\max}$), denote them by $Q_i, i = 1, \cdots, 5, 6, \cdots, 10$. Note that $Q_1 = Q_{\min}$ and $Q_6 = Q_{\max}$. Mathematically,

$$\begin{aligned} Q_i &= \lambda^{(i-1)}[Q_{\min}], i = 1,2,3,4,5 \\ Q_i &= \lambda^{(i-6)}[Q_{\max}], i = 6,7,8,9,10 \end{aligned} \tag{18}$$

Mathematically, step 1 is creating 200 (2N in general) string cycles according to Proposition 13**,** which paves the way toward the global optimum searching.

**Step 2: Rank the strings.** Fitness checking and best-worst string vectors creation. It is divided three sub-steps:

(1) Combine $Q_i, i = 1, 2, \cdots, 10$ into a super string vector, denote by $Q$.

(2) Sort the 1000 strings in $Q$ by ascending order according to objective function values with respect to the 1000 strings, and denote the ranked string vectors as $Q'$.

(3) Define the top 100 strings of $Q'$ as $Q'_{\min}$ and the bottom 100 strings of $Q'$ but reverse them in descending order as $Q'_{\max}$.

Mathematically, Step 2 is utilizing the 200 cycles of objective function values in which 200 minimum candidate solutions and maximum candidate solutions are constructed according to Proposition 15.

**Step 3: Best element select and worst element removes.** Intuitively, this step utilizes genetic engineering ideas: for seeking the best healthy gene combinations it is necessary to keep the best individual gene in the particular position within the gene sequence and also remove the worst individual gene from the particular position within the gene sequence. What we will act is just an imitation to gene selecting and removing in the five-element string sequences created in Step 2, i.e., $Q'_{\min}$ and $Q'_{\max}$ in terms of $\lambda[\ ]$ operation. This is divided into two sub-steps.

(1) **Packed-Rolling operation.** This sub-step performs operations within $Q'_{\min}$ and $Q'_{\max}$ respectively.

If we aim tat search global minimum of the given objective function, strings in $Q'_{\max}$ will be regarded as worse gene sequences and thus the first string corresponding to the maximum objective function value is the worst one. Similarly, strings in $Q'_{\max}$ will be regarded as better gene sequences and thus the first string corresponding to the minimum objective function value is the best one.

The Matlab pseudo-code of packed rolling operation is listed as follows.

Assume ranked candidate solutions denote as matrix $Q$, the matrix size is row multiply column.

```
for i=1:1: row-4
for j=1:1: column
if Q(i,j)== Q(i+1,j) && Q(i,j)~=4
Q(i+1,j)=Q(i+1,j)+1;
elseif Q(i,j)== Q(i+1,j) && Q(i,j)==4
Q(i+1,j)=0;
elseif Q(i,j)== Q(i+2,j) && Q(i,j)~=4
Q(i+2,j)= Q(i+2,j)+1;
elseif Q(i,j)== Q(i+2,j) &&Q(i,j)==4
Q(i+2,j)=0;
elseif Q(i,j)== Q(i+3,j) && Q(i,j)~=4
Q(i+3,j)= Q(i+3,j)+1;
elseif Q(i,j)== Q(i+3,j) && Q(i,j)==4
Q(i+3,j)=0;
elseif Q(i,j)== Q(i+4,j) && Q(i,j)~=4
Q(i+4,j)= Q(i+4,j)+1;
elseif Q(i,j)== Q(i+4,j) && Q(i,j)==4
Q(i+4,j)=0;
end
end
end
```



Verbally, Packed-Rolling operation can explained as follows: Defined five strings as a “package”, within the selected package, the best string is the first item of the package. Then examining the first element (location) in the second string, if the element repeats the first element of the best string, $\lambda[\ ]$ operator should be applied to the repeated element one-time. Next, the second element (position) of the second string is examined, if it repeats the second element of the best string, $\lambda[\ ]$ operator should be applied. Keep on the checking every individual element of the second string until the last one (position). Repeat the check and replacement operations with respect to the third, fourth and fifth string in the package. Then we select the second package, in which the second string in the string vector $Q$ will be defined as the first string of this package. Perform the check and replacement operations within the second package until finished. Then the third package is defined where the third string in the string vector $Q$, and perform the check and replacement operations within the third package, and so on until the ROW-4$^{th}$ package is defined and checked. At the end of Packed Rolling, all the strings are re-evaluated via $f\left(\underline{u}_{\min}+u_r O \underline{e}\right)$, and accordingly re-ranked in ascending order for forming new string vector $Q_{\min}$ and in descending order for $Q_{\max}$.

(2) **Excise worst elements.** Different from Packed-Rolling sub-step, this operation is performed by comparing the corresponding elements between $Q_{\min}$ and $Q_{\max}$. Intuitively, excising the worst elements with respect to the best strings from the opposite string vector is similar to excising bad gene from the gene sequence by comparing to a healthy gene sequence.

In this sub-step, two corresponding strings (candidate solutions) from $Q_{\min}$ and $Q_{\max}$ each are selected and compare their corresponding elements sequentially. If we are seeking global minimum, then the strings from $Q_{\max}$ will be "sick" ones while the strings from $Q_{\min}$ will be regarded as "healthier" ones. For the same location, if the healthier string contains element being the same as the element at the same location in the "sick" string, this individual element at this location should be excised and replaced by the element at the same location from the best string. (i.e., the first string in $Q_{\min}$).

The pseudo-code of Matlab describes how to excise unhealthy elements from relevant the strings. Assume string vector $Q$ is for generating global minimum, and string vector $Q1$ is for generating global maximum.

```
for i=1:1:row-1
for j=1:1:column
if Q1(1,j)== Q(i+1,j)
Q(i+1,j)= Q(1,j);
end
if Q(1,j)== Q1(i+1,j)
Q1(i+1,j)= Q1(1,j);
end
end
end
```

In the excising operation, the first strings in $Q$ and $Q1$ are defined as the best elements and the worst elements respectively. If for a given location the element in $Q$ repeats the element at the same location in $Q1$, this particular element should be excised and replaced by the element at the same location of the first string in $Q$.

At the beginning of scheme running, the excising operation might cause the convergence too quick (such that trap into local optimum), and during the whole algorithm running period, it also might cause some healthy elements been excised. However Proposition 15 guarantees the success of the scheme as what we pointed in Remark 16.

At the end of Step 3, new string vector $Q_{\min}$ in ascending order and $Q''_{\max}$ in descending order will be generated. The flow chart of the naïve string optimization searching scheme is shown in Figure 3.1.3.

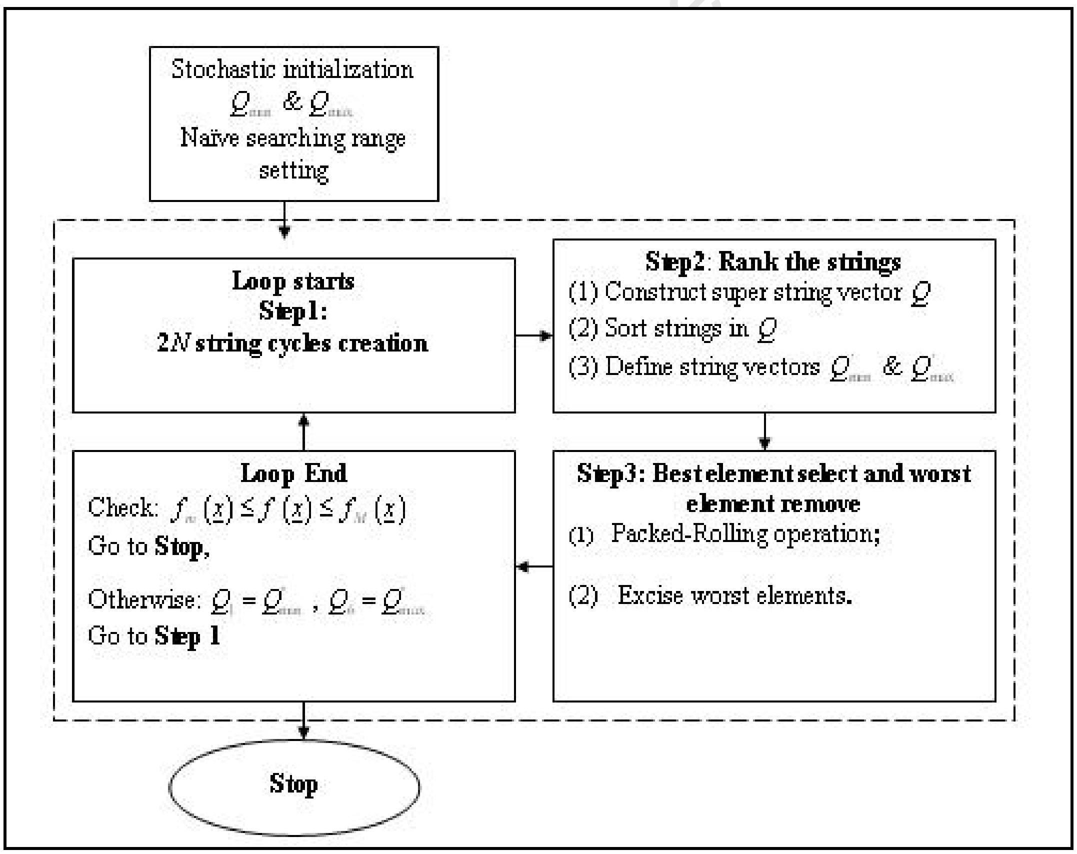



Figure 3.1.3 Flow chart of naïve five-element string algorithm

The naïve five-element string algorithm can be stated as following:

| |
|---|
| Initialization (generating string population $Q_{\min}$ and $Q_{\max}$ stochastically).<br>Start loop<br>1. $2N$ string cycles creation;<br>2. Rank the strings;<br>3. Best element select and worst element remove;<br>4. Check the loop stop criteria: (yes, GoTO 1, yes,<br>Loop Stops);<br>End loop |

### 3.1.7 Illustrative examples

We use five-element naïve string algorithm to search the global optimum for three objective functions: Rosenbrock function, Rastrigin function and Griewank function. Also, we use GA performing the three functions as comparison.

**A. Rosenbrock function**

$$f(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2 \tag{19}$$

The naïve string algorithm searching by 52 loops gives $f_{\min} = 0.0015$, and the global minimum state $\left(x_1^m, x_2^m\right) = (1.0117, 1.0199)$. The searching area is $\mathbb{D} \triangleq \left[-10^6, 10^6\right] \times \left[-10^6, 10^6\right]$.

**B. Rastrigin function**

$$f(x_1, x_2) = 2 + x_1^2 - \cos(18x_1) - \cos(18x_2) \tag{20}$$

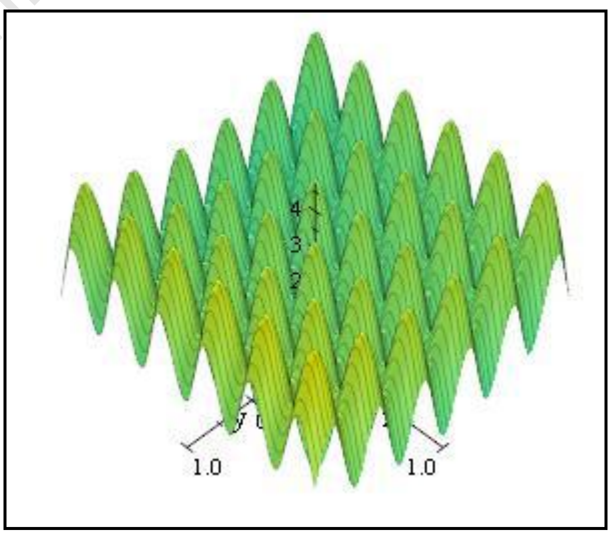



Figure 3.1.4 3D-plot of Rastrigin function

The global minimum is 0 at $\left(x_1^m, x_2^m\right) = (0,0)$ and it is well-known that in area $[-1,1]\times[-1,1]$ there are more than 50 local minima spreading as a lattice around the global minimum.

The naïve string algorithm searching in the area $\mathbb{D} \triangleq \left[-10^6,10^6\right]\times\left[-10^6,10^6\right]$ by 26 loops gives the global minimum 0 at $\left(x_1^m, x_2^m\right) = (0,0)$.

**C. Griewank function**

This function is 10-dimensional. In the cube $[-600,600]^{10}$, there are thousands of local minima around and the global minimum 0 at the origin.

Using naïve string algorithm to search in the cube $\mathbb{D} \triangleq \left[-10^6,10^6\right]^{10}$, by 77 loops, the algorithm locates $x_1^m,\cdots,x_{10}^m = -0.0041,\cdots,-0.0041$ which gives global minimum 0.00010846.

Table 3.1.2 Comparisons between GA and five-element naïve string algorithm

| Function | algorithm | Searching Cube | Loops | Global min |
|---|---|---|---|---|
| (1) | GA | $\left[-10^6,10^6\right]^2$ | 57 | 0.11935 |
| | NSA | $\left[-10^6,10^6\right]^2$ | 52 | 0.0015 |
| (2) | GA | $\left[-10^6,10^6\right]^2$ | 51 | 1.2178 |
| | NSA | $\left[-10^6,10^6\right]^2$ | 26 | 0.00 |
| (3) | GA | $\left[-10^6,10^6\right]^{10}$ | 52 | 0.12506 |
| | NSA | $\left[-10^6,10^6\right]^{10}$ | 77 | 0.0001085 |
| (3)* | GA | $-600,600^{10}$ | 35 | 0.3243650 |
| | NSA | $-600,600^{10}$ | 7 | 0.00001085 |

## 3.1.8 Concluding remarks

It is quite promising that the naïve five-element string algorithm has demonstrated its excellent global searching capability with competitive speed (measured by loop number) and competitive quality (in terms of the global minimum). The naïve sting algorithm offers global minimum and maximum at the same time. It is also exciting that when the search "cube" is reduced, the searching loops decreases greatly and the search quality increases without any doubts. However, the “reduced” search cube implies a constrained optimization or more information is required for the objective function. Such a demand is often impossible to be satisfied in GA modelling exercises.

The algorithm has three fundamental features:

1. The states of the system is represented by strings of 5 elements {0,1,2,3,4} and hence the search of the optimal state(s) is realized by string manipulations;
2. A weighting system is created for a balanced global and local search to avoid the scheme trapping in local optimum;
3. The string operation is a pseudo-linear transformation, which involves if-else logical operator, such that the searching the optimum of a nonlinear multivariate objective function is essentially linear.

Finally, there is a trend in scientific research–complication. It is true that real world is complicated. However, any complicated phenomenon can be decomposed into simple ones. It is fair to say that to pursue simple one, rather, complicated should be the basic goal of scientists. Our naïve five-element string algorithm is the simplest one with high efficiency and worth to be promoted.

# 3.2 Lambda Algorithm

Yanhong Cui, Renkuan Guo

Department of Statistical Sciences, University of Cape Town

Private Bag, Rhodes' Gift, Rondebosch 7701, Cape Town, South Africa

hadisna@hotmail.com renkuan.guo@uct.ac.za

Danni Guo

South African National Biodiversity Institute,

Private Bag X7, Claremont 7735, Cape Town, South Africa

d.guo@sanbi.org.za



This paper has been published in the Journal of Uncertain Systems, Vol. 4, No. 1, pp. 22-33. 2010.

ISSN: 1752-8909 (PRINT), ISSN: 1752-8917 (ONLINE), Publisher: World Academic Union.

### 3.2.1 Abstract

In this paper, we propose a new global optimization algorithm inspired by stochastic model and graph theory, which is named as lambda algorithm (LA). The new algorithm utilizes strings of element from member set {0, 1, 2, 3, 4} to represent the values of candidate solutions (typically represented as vectors in n-dimensional Euclidean space). LA draws useful information from both repeated and unrepeated elements from candidate solutions (strings), to simulate best global schema towards final optimization. Except the mathematical operations for evaluating the objective function, sort procedure, creating initial population randomly, the algorithm only involves if-else logical operation. In contrast to existing global optimization algorithms, the lambda algorithm engages the simplest mathematics but reaches the highest searching efficiency.

**Keywords:** string, λ operator, weight matrix, genetic algorithm, five-element string algorithm

### 3.2.2 Introduction

Optimization is one of challenging and active mathematical research branches. Particularly, the topic of global optimization is of critical importance because of the high demand and wide applications in science, business, economy, and industry.

We have created a new global optimization search algorithm with following features:

1. First, the algorithm use lambda operator to build a five-state regular Markov chain model and initialize its individual solution as strings of 0s, 1s, 2s, 3s, and 4s. Let P be a regular transition probability matrix with state space S= {0, 1, 2, 3, 4}. Then the limiting probability distribution $\Pi = (\pi 0, \pi 1, \pi 2, \pi 3, \pi 4)$ with $\pi i = 0.2$ and satisfying $\Pi = \Pi P$. The property of limiting probability will hold the algorithm search result being independent of initial individual solution setting, keeping algorithm to seek the intrinsic harmonious state of a system under investigation.
2. Secondly, the algorithm considers five-state system as a filter, filtering advanced schema (ex.10**243**2) (Genetic algorithm's idea) from each individual solution. The key issue is selecting 3 different candidate solutions, their fitness values are similar. The algorithm made each of them compare to each other, then drawing out information from repeated and unrepeated digits. When most of candidate solutions of population have the same digit at

same position, the probability of the digit portion is larger than a confidence value $\gamma$, $0 \le \gamma \le 1$, thus the digit at this position goes to a steady state.

3. Thirdly, the algorithm operation is a pseudo-linear transformation such that searching the optimum of a nonlinear multivariate objective function is essentially linear. According to the limiting probability distribution property and advanced schema filtering, a weighting system is created, which leads some large weighted digits of population soon approach a probability stationary. Then extracting the steady digits by shrink the search area of candidate solutions, this procedure will repeat again and again since the algorithm is running, the algorithm will run towards to final optimum direction until the search area be shrink smaller enough. Thus, Cauchy sequences will be generated in this algorithm to satisfy the optimum approach condition.

### 3.2.3 An inspiring example

The objective function for illustrating purpose is the Rosenbrock function

$$f(x,y) = 100\left(y - x^2\right)^2 + \left(1 - x\right)^2 \tag{1}$$

The global minimal value is 0 of Rosenbrock function at (x,y)=(1,1). The plot of (1) in Figure 3.2.1 offers an intuitive view on the features optimality of (1).

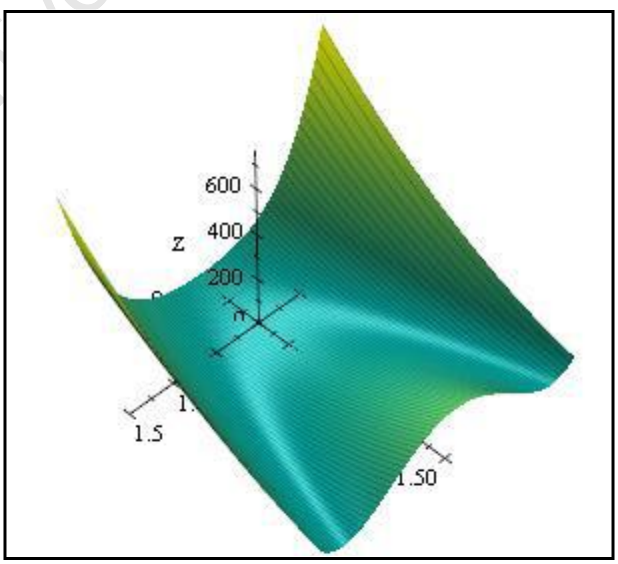



Figure 3.2.1 Plot of Rosenbrock function

We used GA (Matlab help Example, 2008) to search the global minimum of Rosenbrok function. However, contrary to the global optimization searched by GA, the "global" minimal value 0.11935 at (x,y)=(0.681,0.477) is reported.

The case of GA's "failure" to search the true global minimum (for given computing time) here inspires us to consider the fundamental weakness of GA. It is noticed that GA, as a global optimization algorithm, differs from many other algorithms.

Let $f\left(\underline{x}\right)$ be the objective function, where $\underline{x}=\left(x_1,x_2\right)^T\in D\subset R^2$. In many optimization searching algorithms, a typical exercise is trying to improve the optimality within the neighbourhood. It is obvious that the increment in $\underline{x}$ approach typically leads to a local optimum.

GA does not work on system state $\underline{x}=\in D\subset R^n$ of the objective function $f\left(\underline{x}\right)$ directly, rather it uses string like 0011001100100111010111100 for representing the state $\underline{x}^T=\left(x_1,x_2,\cdots,x_n\right)$ and hence may possess better global coverage. However, GA string member set is $\{0,1\}$. Inevitably, the change in string may not change the state $\underline{x}^T=\left(x_1,x_2,\cdots,x_n\right)$ efficiently for covering the whole domain because the element change in a string is 1.

GA string member set {0,1} is too simple, which restrict itself not being able to operate complicated "mutation", only after "selection" and "crossover" two procedures' support, which then simulates advanced schema (ex.10*01**1). But for "crossover", for now we only follow "random crossover" or "60% crossover". It is a fuzzy intuitive way to operate, which restricts us to improve GA itself in further research. So, by expanding the string member set, we can make a more complicated "mutation" operation, and drop the fuzzy "crossover" idea, and it may lead to a better way to generate a more efficient heuristic algorithm.

### 3.2.4 String representation of the state of objective function

An improvement strategy is to expand string member set. Now let us formally establish the string representation related concepts.

**Definition 1:** A string is a sequence of integers; denote by $n_1,n_2,\cdots,n_t$, Number $P$ is called the length of a string. For operational convenience, a string may be repressed by a row vector, $\underline{n}^T=\left(n_1,n_2,\cdots,n_p\right)$.

**Definition 2:** The collection of the elements for constructing a string, denoted by {0,1,…,s−1}, is termed as an element set for a string. S is called the size of the element set of a string (i.e., the number of elements in the element set).

**Conjecture 3:** The size of the element set of a string used in a lambda algorithm is a prime number. In GA, the size of the element set {0, 1} is prime number 2. Prime number 3, 5, 7, 11, etc can also be used. If the size of the element set is 7, then the element set is {0, 1, 2, 3, 4, 5, 6}.The length of a string P should be at least n(s+1).

**Definition 4:** Let $\underline{x}=\left(x_1,x_2,\cdots,x_n\right)^T \in D \subset R^2$ denote system state, which is also representing the candidate solution. Then the length of the string representing $\underline{x}$ is $p>nu$ if the size of the string element set is $S$ , $u>s$ , $u$ is called the basic unit size of a string. The string representation for $\left(x_1,x_2,\cdots,x_n\right)^T$ is

$$e_1e_2\cdots e_ue_{u+1}\cdots e_{2u}\cdots e_{(n-1)u+1}\cdots e_{nu} \tag{2}$$

An intuitive correspondence between the state $\underline{x}$ and the representing string is

$$\underbrace{e_1e_2\cdots e_u}_{x_1}\,\underbrace{e_{u+1}\cdots e_{2u}}_{x_2}\cdots\underbrace{e_{(n-1)u+1}\cdots e_{nu}}_{x_n} \tag{3}$$

**Lemma 5:** Let the system state be $\underline{x}\in D\subset R^n$ , and $\underline{e}$ be a string representation (of the system state) with element set size $S$ and string length $n\left(S+1\right)$. Let $u_r=\max\left\{u_{\max,i}-u_{\min,i}\right\}$,

where $\underline{u}_{\min}\le\underline{x}\in D$, $\underline{u}_{\max}\ge\underline{x}\in D$. The weight matrix $O=\left(o_{ij}\right)_{n\times nu}$ with the $i^{th}$ row vector $\underline{o}_i^T$ having a form

$$\left(0,0,\cdots 0,\cdots,\frac{S^s}{S^{s+1}},\frac{S^{s-1}}{S^{s+1}},\cdots,\frac{S^0}{S^{s+1}},\cdots,0,0,\cdots,0\right) \tag{4}$$

where the nonzero weights are located at the $i^{th}$ segment. Then the system state is a linear transformation of the $S$ -element string representation

$$\underline{x}=\underline{u}_{\min}+u_rO\underline{e} \tag{5}$$

**Definition 6:** If e is an element of a string with element set {0, 1, 2, 3… s−1}, then the value changing rule is

$$e\begin{cases} e+1 & \text{if } e\in\{0,1,2,\cdots,s-2\} \\ 0 & \text{if } \quad e=s-1 \end{cases} \tag{6}$$

Definition 6's element $e$ value change rule is called λ lambda operation, and λ[ ] is called lambda operator, we will discuss and define that part carefully at later section. In the remaining sections of this paper, we will use 5-element string for illustration and the establishment of the lambda algorithm.

### 3.2.5 Five-element string representation

For clarity, we will use numerical examples for illustrating the necessity and advantages of string representation.

**Example 7:** Let $D=[u_{\min},u_{\max}]\times[u_{\min},u_{\max}]$ be the domain for an objective function $f(x_1,x_2)$. Assume that a string 1 2 4 3 0 1 2 4 3 2 1 1 represents $(x_1,x_2)$: the first 6 elements, i.e., 1 2 4 3 0 1, in the string stand as $x_1$ and the second 6 elements, i.e., 2 4 3 2 11, stand as $x_2$. The element set is {0, 1, 2, 3, 4}, the size of element set is 5, the basic unit u=5+1=6. The length of the string 1 2 4 3 0 1 2 4 3 2 11 is 2u=12, which is the number of units occupied in computer.

Mathematically, the linear system linking the five-element string and the system state can be expressed by

$$\begin{cases} x_1 = u_{\min} + (u_{\max} - u_{\min})\sum_{j=1}^{6} e_j \dfrac{5^{6-j}}{5^6} \\ x_2 = u_{\min} + (u_{\max} - u_{\min})\sum_{j=7}^{12} e_j \dfrac{5^{12-j}}{5^6} \end{cases} \tag{7}$$

Let

$$O_{2\times 2u} = \begin{bmatrix} \dfrac{5^5}{5^6} & \cdots & \dfrac{5^0}{5^6} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \dfrac{5^5}{5^s} & \cdots & \dfrac{5^0}{5^s} \end{bmatrix},\ \underline{e}_{2u\times 1} = \begin{bmatrix} e_1 \\ \vdots \\ e_6 \\ e_7 \\ \vdots \\ e_{12} \end{bmatrix} \tag{8}$$

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},\ \underline{u}_{\min} = \begin{bmatrix} u_{\min} \\ u_{\min} \end{bmatrix},\ u_r = u_{\max} - u_{\min}$$

Then a matrix equation for string to state vector transformation is

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} u_{\min} \\ u_{\min} \end{bmatrix} + \left(u_{\max} - u_{\min}\right) \begin{bmatrix} \frac{5^5}{5^6} & \cdots & \frac{5^0}{5^6} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \frac{5^5}{5^6} & \cdots & \frac{5^0}{5^6} \end{bmatrix} \begin{bmatrix} e_1 \\ \vdots \\ e_6 \\ e_7 \\ \vdots \\ e_{12} \end{bmatrix} \tag{9}$$

Matrix O is actually a weighting system which promotes the changes in $\left(x_1, x_2\right)$ according to the location of an individual member in the string as well as the changing size of the member.

In other words, the mechanism underlying the usage of string lies on that the weighting system, i.e., $\left\{\frac{5^5}{5^6}, \frac{5^4}{5^6}, \frac{5^3}{5^6}, \frac{5^2}{5^6}, \frac{5^1}{5^6}, \frac{5^0}{5^6}, 0,0,0,0,0,0\right\}$, assigned to the 6 members in the first half of the string and $\left\{0,0,0,0,0,0, \frac{5^5}{5^6}, \frac{5^4}{5^6}, \frac{5^3}{5^6}, \frac{5^2}{5^6}, \frac{5^1}{5^6}, \frac{5^0}{5^6}\right\}$, the weighting system assigned to the 6 members in the second half of the string create the possibility that change in the member of the string will have different impacts.

A string, denoted by $e_1 e_2 \cdots e_6 e_7 e_8 \cdots e_{12}$, the 1-6 members are the first half of the string, representing $x_1$, the 7-12 members are the second half of the string, representing $x_2$. Logically, changes in $e_1$ and $e_7$ will result in largest changes in $x_1$ and $x_2$ respectively, because the highest weight 0.2 is assigned to them, while changes in $e_6$ and $e_{12}$ will result in the smallest changes in $x_1$ and $x_2$ respectively, because the lowest weight 0.000064 is assigned to them. Therefore, a well-constructed string element change scheme will have a balanced global searching capability as well as local fine-tune capacity.

**Example 8:** (Continued) Define $u_{\min} = -10^{10}$, $u_{\max} = +10^{10}$, then $u_r = u_{\max} - u_{\min} = 2 \times 10^{10}$. String 1: 1 2 4 3 0 1 2 4 3 2 11 used in Example 7 is the base for observing the impacts from string member changes. String 2 changes the first element of the String 1 by adding 1 and the seventh element of the String 1 by adding 1, which is the smallest shift in size at highest weight 0.2. The change in $x_1$ and $x_2$ is quite large with distance 5656854249.5. However, String 3 changes the sixth element of the String 1 by adding 3 and the seventh element of the String 1 by adding 3, which is the largest shift in size at highest weight 0.000064. The change in $x_1$ and $x_2$ is much small with distance 202276452.4. Table 3.2.1 summaries the changes and impacts.

Table 3.2.1 The impacts of weights in global searching and local tune-up

| String | x1 | x1 | \|△x\| |
|---|---|---|---|
| 1 2 4 3 0 1<br>2 4 3 2 1 1 | -3662720000 | 175680000 | |
| 2 2 4 3 0 1<br>3 4 3 2 1 1 | 337280000 | 5751680000 | 5656854249.5 |
| 1 2 4 3 0 4<br>2 4 3 2 1 4 | -3460480000 | 1755520000 | 202276452.4 |

It is important to emphasize here that the value of a string depends on three factors: (1) value of individual element in a string from {0, 1, 2, 3, 4}; (2) the location (or position) of a specific element $e_i$ ; (3) the combination of all elements appeared in the given string. Formally, let us define the five-element if-else operator, called as $\lambda$ operator.

### 3.2.6 Lambda operation

Before we introduce lambda operator, let us first review some useful knowledge of stochastic process and graph theory.

Suppose the Markov chain $M$ is a digraph (directed graph) $G=G_M$ having the set of nodes $N=\{1,2,\ldots,n\}$ and the set of edges $E$. Each node corresponds to a state of $M$, and $G$ contains edge $(i,j)\in E$ if and only if $p_{ij}>0$. Thus the digraph, or state transition diagram, $G$ captures the structure of possible one-step state transitions. For any state $i$ we let $p_i$ denote the probability that, starting in state $i$, the process will ever re-enter state $i$. State $i$ is said to be recurrent if $p_i=1$ and transient if $p_i<1$.In graph-theoretic terms, $p_{ij}^k > 0$ means there is a directed path $Q$ of length $l(Q)=k$ (number of edges) from node $i$ to node $j$ in $G$. If this holds for some $k\geq 0$, then node $j$ is accessible from node $i$, written $i\rightarrow j$. If both $i\rightarrow j$ and $j\rightarrow i$ hold, then we say that states $i$ and $j$ communicate, written $i\leftrightarrow j$. A path joining a node to it is called a circuit. If this circuit contains no repeated nodes, then it is a cycle.

**Theorem 9:** If G is strongly connected then there is a unique stationary distribution $\pi$ for M. Moreover, this distribution satisfies $\pi_j > 0$ for all $j \in N$ .

**Definition 10:** (λ operator) Let $e \in \{0,1,2,3,4\}$, then

$$\lambda[e]=\begin{cases} e+1 & if \quad e \in \{0,1,2,3\} \\ 0 & if \quad e=4 \end{cases} \tag{10}$$

$\lambda^{(l)}[\ ]$ is $l$th order λ operator , which repeats λ operation m times.

**Definition 11:** (Modulo operator) Let $d$ be a positive integer, $q$ be the quotient and $r$ remainder $r$ satisfying

$$d = nq + r \tag{11}$$

Then we write the modulo operation as

$$d \mod(q) = r \tag{12}$$

**Definition 12:** Let $e=(e_1,e_2,\cdots,e_g)$ be a five-element string, then the λ operation on a string is a component-wise operation, i.e.

$$\lambda\left[\underline{e}\right] = \left(\lambda\left[e_1\right],\lambda\left[e_2\right],\cdots,\lambda\left[e_g\right]\right) \tag{13}$$

Furthermore, let $A=(e_{ij})_{h\times g}$ be a five-element matrix, i.e., $e_{ij} \in \{0,1,2,3,4\}$, then

$$\lambda[A]=\left(\lambda\left[e_{ij}\right]\right)_{h\times g} \tag{14}$$

**Proposition 13:** $\lambda^{(l)}[e]=\lambda^{(l \mod(4))}[e]$, where $\lambda^{(0)}[e]\equiv e$.

**Proof:** Note that $e \in \{0,1,2,3,4\}$, the number e only has five choices. For example, $e=0$,

$$\lambda^{(5)}[e]=\lambda^{(4)}[1]=\lambda^{(3)}[2]=\lambda^{(2)}[3]=\lambda^{(1)}[4]=0 \tag{15}$$

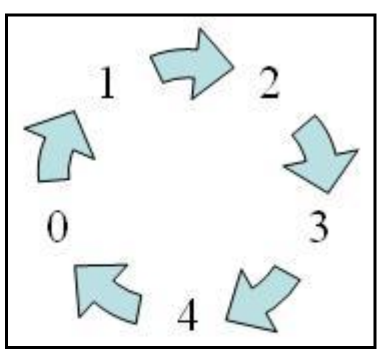



Figure 3.2.2 $\lambda[\ ]$ operation cycle

For any element $e \in \{0, 1, 2, 3, 4\}$, one-time λ [ ] operation shifts the element $e$ from current position into the next 1st position along the cycle shown in Figure 3.2.2. Hence l-time λ [ ] operation shifts the element $e$ from current position into the next lth-position along the cycle. Further, due to the fact that five-element member set {0, 1, 2, 3, 4} only has five members in it, the period of the cycle is 5. Therefore, $\lambda^{(l)}[e] = \lambda^{(l \mod(4))}[e]$ since 0 is the first member of the element set.

**Proposition 14:** For any given five-element string $\underline{e}$, the five-time λ[ ] operated strings form a string cycle. In other words, $\left\{\underline{e}, \lambda^{(1)}[\underline{e}], \lambda^{(2)}[\underline{e}], \lambda^{(3)}[\underline{e}], \lambda^{(4)}[\underline{e}],\right\}$ is a string cycle.

**Definition 15:** (λ spreading operation) Let

$$\begin{aligned} &\underline{x}^{(k)} = \underline{u}_{\min} + u_r O \lambda^{(k)}[\underline{e}], \\ &k = 0,1,2,3,4 \end{aligned} \tag{16}$$

be the corresponding system state of $\lambda^{(k)}[\underline{e}]$. Then $\left\{\underline{x}, \underline{x}^{(1)}, \underline{x}^{(2)}, \underline{x}^{(3)}, \underline{x}^{(4)}\right\}$ is the system state cycle and $\left\{f(\underline{x}), f\left(\underline{x}^{(1)}\right), f\left(\underline{x}^{(2)}\right), f\left(\underline{x}^{(3)}\right), f\left(\underline{x}^{(4)}\right)\right\}$ is the objective function value cycle respect to the string cycle $\left\{\underline{e}, \lambda^{(1)}[\underline{e}], \lambda^{(2)}[\underline{e}], \lambda^{(3)}[\underline{e}], \lambda^{(4)}[\underline{e}]\right\}$, the procedure to generate the string cycle, we called $\lambda$ spreading operation.

**Definition 16:** (λ comparing operation) If we compare 2 strings $\underline{e}_1, \underline{e}_2$, and string $\underline{e}_1$ (candidate solution)'s fitness value better than string $\underline{e}_2$'s, and then we manage some change to $e_2$. $l$ is length of strings $\underline{e}_1, \underline{e}_2$. $e_{1i}, e_{2i}$ is one of the element of $\underline{e}_1, \underline{e}_2$ respectively.

```
For i=1:1:l
If  e_1i = e_2i
e_2i = λ(e_2i)
Else if e_1i ≠ e_2i
e_2i = e_2i
End
End
```

From Definition 16 we knew, compare to $\underline{e_1}$, $\underline{e_2}$ has $p(e_{1i} = e_{2i}) = 0.2$, and $p(e_{1i} \neq e_{2i}) = 0.8$, it means $e_{2i}$ has probability 0.2 equal to $\lambda(e_{2i})$ to change its value, has probability 0.8 to keep its value.

We could consider $\lambda$ comparing operation as a Markov chain model $M$, state space $S = (0,1,2,3,4)$, the transition probability is $P$.

$$P = \begin{vmatrix} 0.8 & 0.2 & 0 & 0 & 0 \\ 0 & 0.8 & 0.2 & 0 & 0 \\ 0 & 0 & 0.8 & 0.2 & 0 \\ 0 & 0 & 0 & 0.8 & 0.2 \\ 0.2 & 0 & 0 & 0 & 0.8 \end{vmatrix} \tag{17}$$

To illustrate the process, we could draw a digraph of M, called graph G [1], [2].

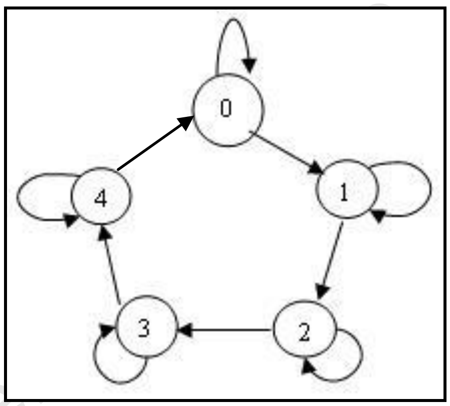



Figure 3.2.3 $\lambda$ comparing operation model diagram

The Figure 3.2.3 graph G is a strongly connected graph, and then there is a unique stationary distribution π for M. (Definition given by theorem 9)

Let $\pi_0, \pi_1, \pi_2, \pi_3, \pi_4$ are stationary probability of state 0, 1, 2, 3, and 4. Then we have

$$\begin{cases} \pi_0 = 0.8\pi_0 + 0.2\pi_4 \\ \pi_1 = 0.8\pi_1 + 0.2\pi_0 \\ \pi_2 = 0.8\pi_2 + 0.2\pi_1 \\ \pi_3 = 0.8\pi_3 + 0.2\pi_2 \\ \pi_4 = 0.8\pi_4 + 0.2\pi_3 \\ \pi_0 + \pi_1 + \pi_2 + \pi_3 + \pi_4 = 1 \end{cases} \tag{18}$$

From Eq. (11) we have:

$$\Rightarrow \pi 0 = \pi 1 = \pi 2 = \pi 3 = \pi 4 = 0.2 \tag{19}$$

And also

$$\text{state } 0 \leftrightarrow \text{state } 1 \leftrightarrow \text{state } 2 \leftrightarrow \text{state } 3 \leftrightarrow \text{state } 4 \tag{20}$$

From the above showing to us, the property of limiting probability will hold the algorithm search result independent of initial individual solutions setting, keep algorithm to seek the intrinsic harmonious state of a system under investigation.

**Proposition 17:** Let

$$\begin{aligned} f_{\min} &= \min\left\{ f\left(\underline{x}\right), f\left(\underline{x}^{(1)}\right), f\left(\underline{x}^{(2)}\right), f\left(\underline{x}^{(3)}\right), f\left(\underline{x}^{(4)}\right) \right\}, \\ f_{\max} &= \max\left\{ f\left(\underline{x}\right), f\left(\underline{x}^{(1)}\right), f\left(\underline{x}^{(2)}\right), f\left(\underline{x}^{(3)}\right), f\left(\underline{x}^{(4)}\right) \right\} \end{aligned} \tag{21}$$

Then the objective function cycle will demonstrate three patterns: (i) $f\left(\underline{x}\right) = f_{\min}$, i.e., the remaining four objective function values are above the cycle starting value $f\left(\underline{x}\right)$; (ii) $f\left(\underline{x}\right) = f_{\max}$, i.e., the remaining four objective function values are below the cycle starting value $f\left(\underline{x}\right)$; (iii) $f_{\min} \le f\left(\underline{x}\right) \le f_{\max}$, i.e., the cycle starting value $f\left(\underline{x}\right)$ falls between cycle minimum and maximum.

**Remark 18:** The weight matrix $O$ in string and system state linking equation $\underline{x} = \underline{u}_{\min} + u_r O \underline{e}$ reveals the ever-changing and controllable character of five element string representation. And the three cycle patterns of objective function values with respect to string cycles reveal that $\lambda[\ ]$ operations guarantee the chance for global optimum searching.

### 3.2.7 LA optimization working principal

From our intuitive observation of whole population of candidate solutions (5-elements represent), the only information we could directly observe is: about 20% digits repeated at same position of each string (candidate solution), and 80% digits are not. Different with many optimization algorithms, the LA draws useful information from both repeated and unrepeated elements from candidate solutions (strings), to simulate best global schema towards final optimization.

Now we start to introduce the optimization principal of LA. Suppose we have N randomly simulated candidate solutions, called population of strings.

1. Rank the whole N strings by fitness checking, sort them by ascending order according to objective function values with respect to N strings.
2. Packed-Rolling operation. The Matlab pseudo-code of packed rolling operation is listed as follows. Assume ranked candidate solutions denote as matrix Q, the matrix size is row multiply column.

```
for i=1:1: row-4
for j=1:1: column
if Q(i,j)== Q(i+1,j) && Q(i,j)~=4
Q(i+1,j)=Q(i+1,j)+1;
elseif Q(i,j)== Q(i+1,j) && Q(i,j)==4
Q(i+1,j)=0;
elseif Q(i,j)== Q(i+2,j) && Q(i,j)~=4
Q(i+2,j)= Q(i+2,j)+1;
elseif Q(i,j)== Q(i+2,j) &&Q(i,j)==4
Q(i+2,j)=0;
end
end
end
```

Verbally, Packed-Rolling operation can be explained as follows: Defined 3 strings as a "package", within the selected package, the best string is the first item of the package. Then examining the first element (location) in the second string, if the element repeats the first element of the best string, $\lambda[\ ]$ operator should be applied to the repeated element one-time. If the second string don't have repeated element, the second element (position) of the second string is examined, if it repeats the second element of the best string, $\lambda[\ ]$ operator should be applied. Then we select the second package, in which the second string in the string vector Q will be defined as the first string of this package. Perform the check and replacement operations within the second package until finished. Then the third package is defined where the third string in the string vector Q, and perform the check and replacement operations within the third package, and so on until the ROW-2nd package is defined and checked.

3. Executive $\lambda$ spreading operation: spreading the string vector Q to $\left\{Q,\lambda^{(1)}[Q],\lambda^{(2)}[Q],\lambda^{(3)}[Q],\lambda^{(4)}[Q]\right\}$, then all the strings is re-evaluated via $f\left(\underline{u}_{\min}+u_r O\underline{e}\right)$.

From $\{Q,\lambda^{(1)}[Q],\lambda^{(2)}[Q],\lambda^{(3)}[Q],\lambda^{(4)}[Q]\}$, then select best fitness N strings as new string vector $Q'$.From Packed-Rolling operation procedure we could see, 3 fitness values most similar candidate solutions be seemed as one "package". Whole population of candidate solutions is classified to ROW-2 packages. The repeated elements according to different classified "package" were transferred to $\lambda^{(4)}[Q]$, unrepeated elements stay at Q. all repeated and unrepeated elements joined with new digits combined to new strings via evaluation. Select 1/5 strings from whole $\{Q,\lambda^{(1)}[Q],\lambda^{(2)}[Q],\lambda^{(3)}[Q],\lambda^{(4)}[Q]\}$, ask comparing two new strings which contains unrepeated and repeated elements. Loop above procedure several times, more and more advanced schema is simulated by testing and selection, successfully draws out the useful information during LA running.

The algorithm operation is a pseudo-linear transformation such that searching the optimum of a nonlinear multivariate objective function is essentially linear. According to the limiting probability distribution property and advanced schema filtering, a weighting system is created via $f(\underline{u}_{\min}+u_r O\underline{e})$, which leads some large weighted digits of population soon approach a probability stationary.

Example: For $\underline{e}=(e_1,e_2,\cdots,e_g)$, by weight comparing, we has $\underline{e}=(e_1>e_2>e_3\cdots>e_g)$, then $e_1,e_2$ might be first and second elements approach probability stationary.

In LA, larger weight element always approach probability stationary more soon than smaller weight element, it means before $e_1$ goes to stationary, $e_2,e_3,\cdots,e_g$ have no way to approach.

**Definition 19:** (Cauchy sequence) A sequence $x_1,x_2,x_3$ of real numbers is called Cauchy, if for every positive real number ε, there is a positive integer N such that for all natural numbers m, n > N, $|x_m-x_n|<\varepsilon$ (where the vertical bars denote the absolute value). (See Figure 3.2.4)

When $e_1$ goes to stationary, the algorithm will ask to extract $e_1$ from $\underline{e}=(e_1,e_2,\cdots,e_g)$. $e_2$ will transform to new $e_1'$, $e_3$ to $e_2'$, …, $e_g$ to $e_{g-1}'$. A new $e_g$ will add by simulation randomly. A new $u_{\max}'$, $u_{\min}'$ will replace $u_{\max}$, $u_{\min}$. Also $|u_{\max}'-u_{\min}'|<|u_{\max}-u_{\min}|$.Repeat this procedure again and again when algorithm is running, until $|u_{\max}^{(n)}-u_{\min}^{(n)}|<\varepsilon$, $\varepsilon$ is a smaller enough positive number, we say the algorithm approach to final optimum.

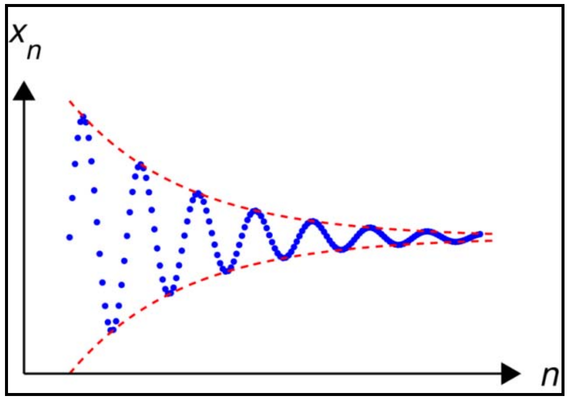



Figure 3.2.4 Cauchy sequence illustration

```
>> [Bestfitness,variables,umax,umin]=J
Please enter the function name(must enter):example @function name
@Rosenbrock
Please enter population size you want(could be empty):default 100

Please enter loop times you want(could be empty):default 100

Please enter String length of each variables(could be empty):default 12

please enter upper bound of variables you want(could be empty):default 10^6

please enter lower bound of variable you want(could be empty):default -10^6

Please enter number of variables(must enter):example 2
2
Please enter Probability measure to control the Process:example 0.5
0.5
Elapsed time is 22.828079 seconds.

Bestfitness =

  3.9144e-020


variables =

    1.0000    1.0000


umax =

    1.0000    1.0000


umin =

    1.0000    1.0000
```

Figure 3.2.5 Two dimensional Rosenbrock LA optimization result

From Figure 3.2.5 we could see, at the beginning, the search area given by program is: $u_{\min} \le x_1, x_2 \le u_{\max}$ $u_{\min} = -1.0 \cdot 10^6$ $u_{\max} = 1.0 \cdot 10^6$ $|u_{\max} - u_{\min}| = 2.0 \cdot 10^6$, after $u'_{\min}$ =0.999999930368000 $u'_{\max}$ =1.000000035225600, $|u'_{\max} - u'_{\min}| = 1.048576 \cdot 10^{-7}$, at computer screen, it shows: $u'_{\min}$ =1.000, $u'_{\max}$ =1.000. Cauchy sequence idea is proved. Confidence probability given by 0.5, if $e_i$ 's portion probability larger than 0.5, we consider $e_i$ goes to stationary and exact $e_i$ out.

## 3.2.8 A lambda algorithm global optimum search scheme



A few terms are defined first. **Stopping time:** The algorithm stops after running for an amount of time in seconds, which is specified as stopping time. **Population size:** The population size defines numbers of rows of matrices, denoted by N. **String length:** The string length defines the number of elements in each five-element string.

**Confidence probability:** Give a probability P, usually let 0.5≤P≤1, if 1st element's portion probability larger than P, then we believe the element will goes to stationary. $n$ : the dimension of objective function. $u_{\min}$ : the lower bound value of input variables. $u_{\max}$ : the upper bound value of input variables.

Before the searching scheme enters algorithm loop the LA nature of the scheme requires the creation of a candidate solution string population. Randomly select numbers from member set {0, 1, 2, 3, 4} uniformly and independently and put them into strings until the string population is established. It is obvious that the discrete uniform random number nature eliminates any possible bias for the starting the algorithm.

**Stochastic initialization:** Randomly generate 2N, say, N=100, five-element strings as candidate solutions, then divide the candidate solutions into two string vectors (two matrices of elements), the first string vector is denoted by $Q_{\min}$ and the second by $Q_{\max}$ . The searching range for the $i^{th}$ component of system state $\underline{x}$ is $[u_{\min}, u_{\max}]$, i.e., $u_{\min} \le x_i \le u_{\max}$ .

**Searching loop:**

**Step 1:** 2N string cycles creation. By applying $\lambda$ spreading operation to $Q_{\min}$ and $Q_{\max}$ respectively, ten string vectors (including $Q_{\min}$ and $Q_{\max}$), denote them by $Q_i$, $i = 1, \cdots, 5, 6, \cdots 10$. Note that $Q_1 = Q_{\min}$ and $Q_6 = Q_{\max}$. Mathematically,

$$\begin{aligned} Q_i &= \lambda^{(i-1)}\left[Q_{\min}\right], i = 1,2,3,4,5 \\ Q_i &= \lambda^{(i-1)}\left[Q_{\max}\right], i = 6,7,8,9,10 \end{aligned} \tag{22}$$

Mathematically, step 1 is creating 200 (2N in general) string cycles according to Proposition 14, which paves the way toward the global optimum searching.

**Step 2:** Rank the strings and checking stationary of elements. Fitness checking and best-worst string vectors creation. It is divided into three sub-steps:

Combine $Q_i, i = 1, 2, \cdots, 10$ into a super string vector, denoted by $Q$.

Sort the 1000 strings in $Q$ by ascending order according to objective function values with respect to the 1000 strings, and denote the ranked string vectors as $Q'$.

Define the top 100 strings of $Q'$ as $Q'_{\min}$ and the bottom 100 strings of $Q'$ but reverse them in descending order as $Q'_{\max}$.

Mathematically, Step 2 is utilizing the 200 cycles of objective function values in which 200 minimum candidate solutions and maximum candidate solutions are constructed according to Proposition 14.

Checking stationary of $Q'_{\min}$. If each variable's string representation, its 1st element's portion probability larger than $P$, then we believe the element will goes to stationary. Exact the element, and a new $u'_{\max}$, $u'_{\min}$ will replace $u_{\max}$, $u_{\min}$. Search area $|u'_{\max} - u'_{\min}| < |u_{\max} - u_{\min}|$. * $Q'_{\max}$ will be selected from $Q_{\min}$ 's λ spreading operation 500 strings. The operation is ranked 500 strings by descending order, $Q'_{\max}$ will be top 100.

**Step 3:** Best element select and worst element remove. Intuitively, this step utilizes genetic engineering ideas: for seeking the best healthy gene combinations it is necessary to keep the best individual gene in the particular position within the gene sequence and also remove the worst individual gene from the particular position within the gene sequence. What we will act is just an imitation to gene selecting and removing in the five-element string sequences created in Step 2, i.e., $Q'_{\min}$ and $Q'_{\max}$ in terms of $\lambda[\ ]$ operation. This is divided into two sub-steps.

(1) **Packed-Rolling operation.** This sub-step performs operations within $Q'_{\min}$ and $Q'_{\max}$ respectively. If we aim at search global minimum of the given objective function, strings in $Q'_{\max}$ will be regarded as worse gene sequences and thus the first string corresponding to the maximum objective function value is the worst one. Similarly, strings in $Q'_{\max}$ will be regarded as better gene sequences and thus the first string corresponding to the minimum objective function value is the best one.

(2) **Excise worst elements.** Different from Packed-Rolling sub-step, this operation is performed by comparing the corresponding elements between $Q_{\min}$ and $Q_{\max}$. Intuitively, excising the worst elements with respect to the best strings from the opposite string vector is similar to excising bad gene from the gene sequence by comparing to a healthy gene sequence. In this sub-step, two corresponding strings (candidate solutions) from $Q_{\min}$ and $Q_{\max}$ each are selected and compare their corresponding elements sequentially. If we are seeking global minimum, then the strings from $Q_{\max}$ will be "sick" ones while the strings from $Q_{\min}$ will be regarded as "healthier" ones. For the same location, if the healthier string contains element being the same as the element at the same location in the "sick" string, this individual element at this location should be excised and replaced by the element at the same location from the best string (i.e., the first string in $Q_{\min}$). The pseudo-code of Matlab describes how to excise unhealthy elements from relevant the strings.

Assume string vector Q is for generating global minimum, and string vector Q1 is for generating global maximum.

```
for i=1:1:row-1
for j=1:1:column
if Q1(1,j)== Q(i+1,j)
Q(i+1,j)= Q(1,j);
end
if Q(1,j)== Q1(i+1,j)
Q1(i+1,j)= Q1(1,j);
end
end
end
```

In the excising operation, the first strings in Q and Q1's are defined as the best elements and the worst elements respectively. If for a given location the element in Q repeats the element at the same location in Q1, this particular element should be excised and replaced by the element at the same location of the first string in Q.

At the beginning of scheme running, the excising operation might cause the convergence too quick (such that trap into local optimum), and during the whole algorithm running period, it also might cause some healthy elements been excised. However Proposition 17 guarantees the success of the scheme as what we pointed in Remark 18.

At the end of Step 3, new string vector $Q_{\min}^{''}$ in ascending order and $Q_{\max}^{''}$ in descending order will be generated. The flow chart of the LA optimization searching scheme is shown in Figure 3.2.6.

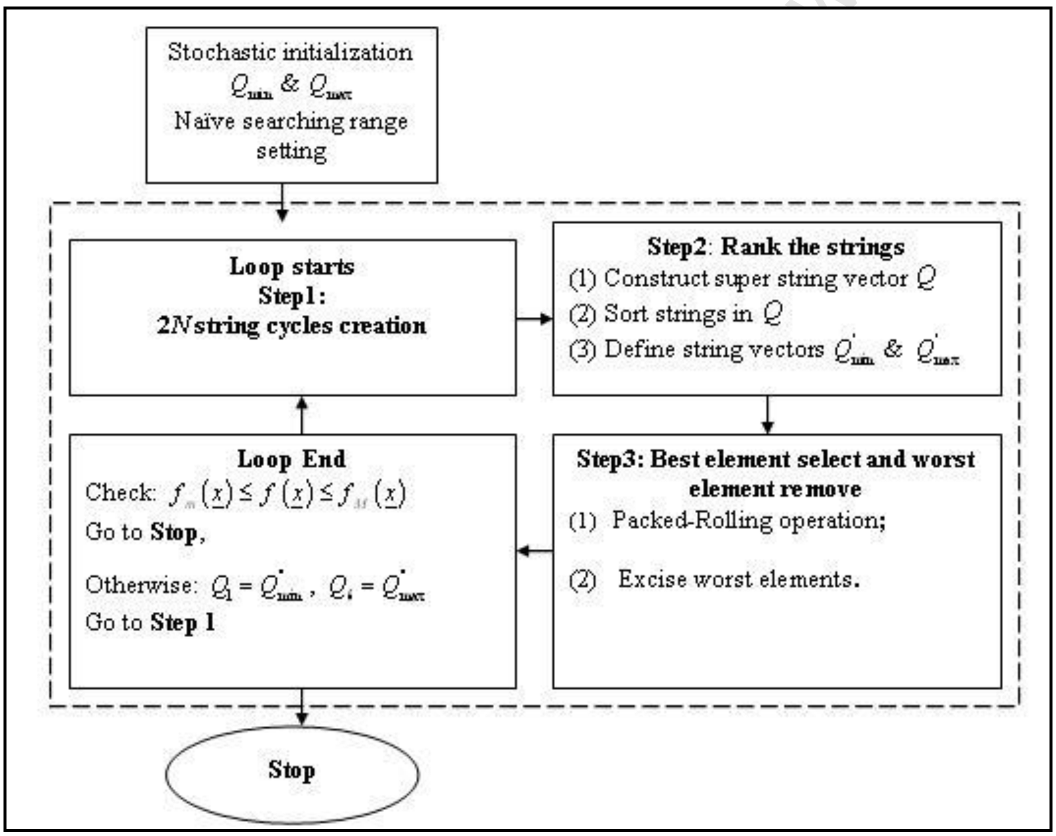



Figure 3.2.6 Flow chart of naïve five-element string algorithm

The LA algorithm can be stated as following:

| |
|---|
| Initialization (generating string population $Q_{\min}$ and $Q_{\max}$ stochastically).<br>Start loop<br>2N string cycles creation; |

Rank the strings and checking stationary;
Best element select and worst element remove;
Check the loop stop criteria: (yes, GoTO 1, yes, Loop Stops);
End loop

## 3.2.9 Illustrative example

We use Lambda algorithm to search the global optimum for three objective functions: Rosenbrok function, Rastrigin function and Griewank function. Also, we use GA performing the three functions as comparison.

**A. Rosenbrok function**

$$f\left(x_1, x_2\right) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2 \qquad (23)$$

The Lambda algorithm searching by 100 loops gives $f_{\min}$ =3.9144e−020, and the global minimum $(x_1^m, x_2^m)$=(0.999999999878608,0. 999999999741594). The searching area is D≡[−10$^6$,10$^6$]× [−10$^6$,10$^6$]. Spend 22.828079 seconds.

**B. Rastrigin function**

$$f \;\; x_1, x_2 \;\; = 2 + x_1^2 - \cos \; 18x_1 \;\; - \cos \; 18x_2 \qquad (24)$$

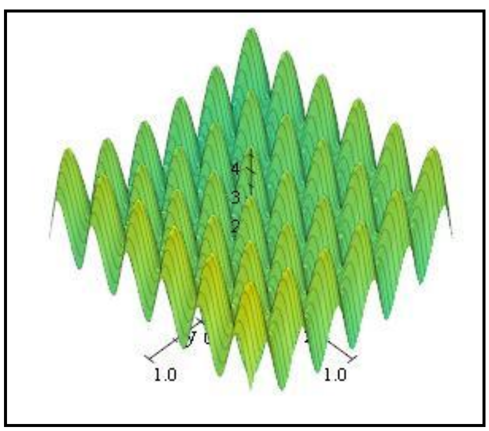



Figure 3.2.7 3D-plot of Rastrigin function

The global minimum is 0 at $(x_1^m, x_2^m)$=(0,0) and it is well-known that in area [−1,1]× [−1,1] there are more than 50 local minima spreading as a lattice around the global minimum.

The naïve string algorithm searching in the area D≡$[-10^6,10^6]\times[-10^6,10^6]$ by 26 loops gives the global minimum 0 at $(x_1^m, x_2^m)$=(0,0).

Table 3.2.2 Comparisons between GA and five-element naïve string algorithm

| Function | algorithm | Searching Cube | Loops | Global min |
|---|---|---|---|---|
| Rosenbrock | GA | $[-10^6,10^6]^2$ | 57 | 0.11935 |
| | LA | $[-10^6,10^6]^2$ | 100 | 3.9144e-020 |
| Rosenbrock | GA | $[-5.12,5.12]^{30}$ | 84 | 105.7045 |
| | LA | $[-5.12,5.12]^{30}$ | 400 | 1.0165 |
| Rastrigin | GA | $[-10^6,10^6]^2$ | 51 | 1.2178 |
| | LA | $[-10^6,10^6]^2$ | 26 | 0.000 |
| Griewank | GA | $[-10^6,10^6]^2$ | 52 | 0.12506 |
| | LA | $[-10^6,10^6]^2$ | 77 | 0.000 |
| Griewank | GA | $[-600,600]^{1000}$ | 52 | 0.55736 |
| | LA | $[-600,600]^{1000}$ | 37 | 2.3360e-011 |

**C. Griewank function**

This function is 1000-dimensional. In the cube $\left[-600,600\right]^{1000}$, there are thousands of local minima around and the global minimum 0 at the origin.

$$f_n(\bar{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, i = 1,2,\cdots,1000; \tag{25}$$

Using lambda algorithm to search in the cube, by 37 loops, the algorithm locates xi=0.2458, i=1,2,…,1000 which gives global minimum 2.3360e-011. Spend 337.4632 seconds.

## 3.2.10 Concluding remarks

It is quite promising that the naïve five-element string algorithm has demonstrated its excellent global searching capability with competitive speed (measured by loop number) and competitive quality (in terms of the global minimum). The naïve sting algorithm offers global minimum and maximum at the same time. It is also exciting that when the search “cube” is reduced, the searching loops decreases greatly and the search quality increases without any doubts. However, the "reduced" search cube implies a constrained optimization or more information is required for the objective function.

In the new algorithm development, (1) The states of the system is represented by strings of 5 elements {0,1,2,3,4} and hence the search of the optimal state(s) is realized by string manipulations; (2) A weighting system is created for a balanced global and local search to avoid the scheme trapping in local optimum; (3) The string operation is a pseudo-linear transformation, which involves if-else logical operator, such that the searching the optimum of a nonlinear multivariate objective function is essentially linear.

Finally, there is a trend in scientific research – complication. It is true that real world is complicated. However, any complicated phenomenon can be decomposed into simple ones. It is fair to say that to pursue simple one, rather, complicated should be the basic goal of scientists. Our naïve five-element string algorithm is the simplest one with high efficiency, and it already has successful applications (Cui, 2008, 2009; Savani and Rao, 2008) should be examined further.

## 3.3 Nash-Lambda Algorithm With Applications In Safety and Reliability



Yanhong Cui, Renkuan Guo

Department of Statistical Sciences, University of Cape Town

Private Bag, Rhodes' Gift, Rondebosch 7701, Cape Town, South Africa

hadisna@hotmail.com renkuan.guo@uct.ac.za

This paper has been published in Summer Safety and Reliability Seminars, SSARS 2011, Journal of Polish Safety and Reliability Association, Vol. 1, pp. 51-58. 2010. ISBN: 978-83-925436-2-6. Editors in Chief: Krzysztof Kolowrocki, Joanna Soszynska and Enrico Zio.

### 3.3.1 Abstract

In this paper, a new algorithm, named as Nash-lambda algorithm by merging Nash equilibrium solution and the lambda algorithm, is proposed. The lambda algorithm, a new global optimization algorithm, is created by imitating ancient Chinese human body system model, which has already demonstrated its simplicity in searching scheme, codes and efficiency in computation comparing to the genetic algorithm. The non-corporative game environments determine the optimization problems which are different from those of the traditional safety and reliability optimizations because of the engagement of the Nash equilibrium for seeking the best strategy. The lambda algorithm serves the searching the Nash equilibrium solution efficiently. In other worlds, the Nash-lambda algorithm is just developed to address the optimization problems of the multiple objective functions representing non-corporative players' interests.



### 3.3.2 Introduction

Safety and reliability optimization problems are a fundamental components intrinsically in the sense that the statistical theory underlying them is built up by a pile of relevant mathematical optimal theories and methodologies. However, since the 911 event occurred in New York, 2001, the threat from the terrorist organizations has merged into the western governments' agenda list (Heymann, 2003; Kaplan et al., 2005; Keohane and Zeckhauser, 2003). Any government or a utility company, say, the electricity power plant, the water supply company, the public transportation network, the international airport, etc. has the responsibility to secure the highest safety and availability to the public, while the terrorist organization wants to destroy or damage the target to the maximum. It is obvious that the players in the game battle are non-corporative. The optimization problem is no longer the traditional one. Nash equilibrium is "a solution concept of a game involving two or more players, in which each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by changing only his own strategy unilaterally". To obtain the solution set of the Nash equilibrium, it is necessary to search it within the players' strategy sets. There have been many search methodologies, for example, Nash-LQ, Nash-polynomial algorithms etc.

It is noticeable that researchers have try to merge Nash equilibrium solution and the genetic algorithm (abbreviated by Nash-GA) for seeking optimal numerical strategies (Sefrioui and

Perlaux, 2000; Spata and Rinaudo, 2010). The lambda algorithm is created by imitating an ancient human body system (Cui et al., 2008, 2009, 2010), also the sister paper in this seminar, "Lambda algorithm and maximum likelihood estimation". In its searching scheme, except the necessary mathematical computations for evaluating the objective function and the creation of the initial "searching population" randomly, the algorithm only involves if-else logical operation and sort procedure. In contrast to existing global optimization algorithms, particularly GA, the lambda algorithm engages the simplest mathematics but reaches the highest searching efficiency. Therefore it is logical to consider in the Nash-GA replacing the genetic algorithm (GA) part by the lambda algorithm for merging Nash equilibrium solution concept with lambda algorithm to achieve the optimal numerical strategies because of the merits of it comparing to GA.

The remaining structure of the paper is stated as following: Section 3.3.3 serves the explanation of the Nash equilibrium solution and related theory; Section 3.3.4 will analyze the merging of Nash equilibrium solution and lambda algorithm and analyzing an numerical example to illustrate the new Nash-lambda algorithm; Section 3.3.5 will discuss briefly the applications in safety and reliability optimizations; Section 3.3.6 concludes this paper.



### 3.3.3 Nash equilibrium solution concept

The game theory is a applied mathematical branch dealing with the behaviour in strategic situations, in which an individual's gain in making choices depends on the choices of the individual's competitors. Game theory studies theory on the rational side of social science in broad sense, including human as well as non-human players e.g., computers, animals, and etc (Wikipedia, 2011).

In n-player non-corporative games, the Nash equilibrium is a solution state, in which an individual player knows the strategies of the others and also knows that no one can gain anything by altering any individual strategy unilaterally while the others keep their strategies unchanged. Such a set of strategy choices and the corresponding payoffs constitute Nash equilibrium (Wikipedia, 2011).

Let $(S, f)$ be a game with $n$ players, in which $S = S_1 \times S_2 \cdots \times S_n$ is the strategy-profile set with the $i^{th}$ player's strategy set $S_i$, $i = 1, 2, \cdots, n$, and $f = f\left(f_1(x), \cdots, f_n(x)\right)$ is the payoff function. When each individual player decides to choose the strategy $x_i$, then a strategy profile $x = \left(x_1, \cdots, x_n\right)$ is obtained so that the $i^{th}$ player $i$ obtains payoff $f_i(x)$. Let $x_{-i}$ be a strategy profile of all players except for the $i^{th}$ player. Note that the payoff depends on the strategy profile chosen, i.e. on the strategy chosen by player $i$ as well as the strategies chosen by all the remaining players.

**Definition 1:** A strategy profile $x^* \in S$ is Nash equilibrium if no unilateral deviation in strategy by any individual player is profitable for that player, that is

$$\forall i, x_i \in S_i, x_i \neq x_i^* : f_i\left(x_i^*, x_{-i}^*\right) \geq f_i\left(x_i, x_{-i}^*\right) \tag{1}$$

A game can have either a pure-strategy or a mixed-strategy Nash Equilibrium, (in the latter a pure strategy is chosen stochastically with a fixed frequency). Nash proved that if we allow mixed strategies, then every game with a finite number of players in which each player can choose from finitely many pure strategies has at least one Nash equilibrium solution.

### 3.3.3.1 Nash equilibrium solution concept

In the multilevel programming problem, the notation level is actually the sets of variables. For example, a bi-level program, (bi-level programming) has two sets of variables.

**Definition 2:** A bi-level program is the optimization problem within which one optimization problem is embedded in another one.

As a matter of fact, the formulation of a bi-level programming problem can be stated simply as:

$$\min_{x \in X, y \in Y} f^u(x, y) \tag{2}$$

Subject to

$$\begin{aligned} &g^u(x, y) \leq 0, \\ &y \in \arg\min_{z \in Y} f^l(x, z) \\ &g^l(x, z) \leq 0, \end{aligned} \tag{3}$$

where

$$\begin{aligned} &f^u, f^l : R^{nx} \times R^{ny} \to R \\ &g^u, g^l : R^{nx} \times R^{ny} \to R^{mu} \\ &X \subseteq R^{nx} \\ &Y \subseteq R^{ny}. \end{aligned} \tag{4}$$

where the variables $z$ are dummy variables.

#### 3.3.3.2 Stackelberg model

Decision making problems in decentralized organizations are often modelled as Stackelberg competitions, which are formulated as two-level mathematical programming problems (Wikipedia, 2011; Masatoshi and Ichiro, 2009; Simaan and Cruz, 1973). Conflict and cooperation among individual players are an essential part of the process. In the Stackelberg game model, there are two kinds of players; the player of the first kind chooses a strategy at the start, and thereafter the player of the second kind with knowledge of the player's strategy of the first kind determines a strategy of the player of the second kind.

In game theory, players are classified as a leader and the remaining ones as the followers. Stackelberg model is a strategic game in which "the leader firm moves first and then the follower firms move sequentially", ...., the constraints for maintaining the Stackelberg equilibrium is that "the leader must know ex ante that the follower observes his action. The follower must have no means of committingto a future non-Stackelberg follower action (Wikipedia, 2011). The Stackelberg model can be solved to find the subgame perfect Nash equilibrium or equilibria (SPNE), i.e. the strategy profile that serves best each player, given the strategies of the other player and that entails every player playing in a Nash equilibrium in every subgame" (Wikipedia, 2011).

**Definition 3:** Let $x \in \mathbb{R}^N$ be partitioned as $x = \left(x^\alpha, x^\beta\right)$, and a compact set $\mathbb{S} \subset \mathbb{R}^N$. Let $f : \mathbb{R}^N \to \mathbb{R}$ and be continuous on $\mathbb{S}$. The set $\Re_f\left(\mathbb{S}\right) \triangleq \left\{ \hat{x} \in \mathbb{S} | f\left(\hat{x}\right) = \max_{x \in \mathbb{S} \cap \left\{x^\beta = \hat{x}^\beta\right\}} f\left(x\right) \right\}$ is the one of rational reactions under function $f$ on the set $\mathbb{S}$.

To formally define the n-player Stackelberg game model, let $x \in \mathbb{R}^N$ be the vector of decision variables for all n players, and let $x$ be partitioned among n players with $x^k \triangleq \left(x_1^k, x_2^k, \cdots, x_{N_k}^k\right) \in \mathbb{R}^{N_k}$, $k = 1, 2, \cdots, n.$ Note that $\sum_{k=1}^n N_k = N$. The game model requires all n players take $x$ from $\mathbb{S}^l$, whose shape determines the ability of the leader player to affect the set of feasible choices of the follow players. Let $f_k : \mathbb{S}^k \to \mathbb{R}$, $k = 1, 2, \cdots, n,$ $\left\{f_1\left(x\right), f_2\left(x\right), \cdots, f_n\left(x\right)\right\}$ the set of continuous functions.

**Definition 4:** Let $x$ be partitioned as $x = \left(x^\alpha, x^\beta\right)$ with $x^\alpha \triangleq \left(x^1, x^2, \cdots, x^{k-1}\right)$ and $x^\beta \triangleq \left(x^k, x^{k+1}, \cdots, x^n\right)$. The level-k feasible region $\mathbb{S}^k \triangleq \Re_{f_{k-1}}\left(\mathbb{S}^{k-1}\right)$ recursively for $k = 2, 3, \cdots, n.$

The set $\mathbb{S}^k$ collects the feasible outcomes resulting from the rational reactions of players at level-i, $i = 1, 2, \cdots, k-1.$. Hence $\mathbb{S}^k$ contains all of the information necessary for player i to evaluate the behaviour of these players. Given the pre-emptive decisions $\left(\hat{x}^{k+1}, \hat{x}^{k+2}, \cdots, \hat{x}^n\right)$ of the first n − k leading players, the optimization problem which must be solved by the player at level k is then

$$\begin{aligned} &\left(L^k\right): \max\ f_k\left(x\right) \\ &\text{s.t.} \\ &x \in \mathbb{S}^k, \\ &x^i = \hat{x}^i,\ i = k+1, \cdots, n \end{aligned} \tag{5}$$

This presents a nested multi-level programming problem.

It is quite obvious that Stackelberg model, a pure strategy optimization may have only one Nash equilibrium, while mixed strategies could have finitely many Nash equilibria (at least one). The lambda algorithm is designed for both pure strategy and mixed strategies optimization for bi-level programming, which is named as Nash-lambda algorithm.

Nash-lambda algorithm allowed program at each loop of optimization evaluate two strategy objective functions. A switch function to decide the rank of all the candidate solutions. If switch=0, then the algorithm according to leader objective function to rank the candidate solutions. If switch=1, then the algorithm according to follower objective function to rank the candidate solutions. $TempF_{leaders}^{best}$, $TempF_{followers}^{best}$ are two variables, which using to record the best optimization result of leader, follower objective function in the elapsed optimization. $\underline{e_{leaders}^{best}}$ is the best fitness string of leader objective function at current loop. $\underline{e_{followers}^{best}}$ is the best fitness string of follower objective function at current loop. $F_{leaders}^{best}$, $F_{leaders}^{judge}$ are fitness values of $\underline{e_{leaders}^{best}}$ from leader, follower objective function evaluation respectively. Similarly, $F_{followers}^{best}$, $F_{followers}^{judge}$ are fitness values of $\underline{e_{followers}^{best}}$ from follower, leader objective function evaluation respectively.

In pure strategy optimization:

| |
|---|
| If $F_{followers}^{best} \geq TempF_{followers}^{best}$, Switch=0<br>Else if $F_{leaders}^{best} \geq TempF_{leaders}^{best}$, Switch=1,<br>End |

The above program code meaning, for leader objective function and follower objective function, each different strategy optimization only allowed jumping once at the algorithm. After one objective function have a better fitness value, and then the algorithm must turn to face another objective to do the optimization. If the algorithm running towards to leader objective function optimization, one selected strings vector $\overline{e_{new}^{first}}$ must let all the candidate solution take the leader variables values given by $\underline{e_{leaders}^{best}}$ . The meaning is, except $\underline{e_{leaders}^{best}}$ , other strings must copy the digits which represent the leader objective function variable $\underline{e_{leaders}^{best}}$ has. Similarly, if the algorithm running towards to follower objective function optimization, one selected strings vector $\overline{e_{new}^{first}}$ must let all the candidate solution take the follower variables values given by $\underline{e_{followers}^{best}}$ .

The optimization result is, after “step by step”, or say one time by one time altering optimization, if one way of the optimization is stopped, which meaning one way of the strategy is successful, a pure strategy reaches the Nash equilibrium.

In mixed strategies optimization:

If $F_{leaders}^{judge} \geq TempF_{followers}^{best}$ , Switch=0
Else if $F_{followers}^{judge} \geq TempF_{leaders}^{best}$ , Switch=1,
End

The above program code meaning, instead of “step by step” altering optimization, the algorithm allowed optimization continues jumping at one direction. Only when the current best fitness is the best fitness of both leader and follower objective function, the algorithm allowed the optimization towards to another way. The optimization result is more balanced in this way, which can give many more Nash equilibrium for different strategies. The flow chart of Nash-lambda algorithm is showing in Figure 3.3.1.

## 3.3.4 A numerical example

In this section, we consider a bi-level programming with free followers in which the leader has a decision vector $(x_1, x_2, x_3)$ and the three followers have decision vectors $(y_{i1}, y_{i2}), i = 1, 2, 3,$ .see (Liu, 1998).

$$
\begin{cases}
\max\limits_{x_1,x_2,x_3} y_{11}^* y_{12}^* \sin x_1 + 2y_{21}^* y_{22}^* \sin x_2 + 3y_{31}^* y_{32}^* \sin x_3 \\
\text{subject to:} \\
x_1 + x_2 + x_3 \le 10, x_1 \ge 0, x_2 \ge 0, x_3 \ge 0, \\
\left(y_{11}^*, y_{12}^*, y_{21}^*, y_{22}^*, y_{31}^* y_{32}^*\right) \text{ solves the problems} \\
\begin{cases}
\max\limits_{y_{11},y_{12}} y_{11} \sin y_{12} + y_{12} \sin y_{11} \\
\text{subject to:} \\
y_{11} + y_{12} \le x_1, y_{11} \ge 0, y_{12} \ge 0
\end{cases} \\
\begin{cases}
\max\limits_{y_{21},y_{22}} y_{21} \sin y_{22} + y_{22} \sin y_{21} \\
\text{subject to:} \\
y_{21} + y_{22} \le x_2, y_{21} \ge 0, y_{22} \ge 0
\end{cases} \\
\begin{cases}
\max\limits_{y_{31},y_{32}} y_{31} \sin y_{32} + y_{32} \sin y_{31} \\
\text{subject to:} \\
y_{31} + y_{32} \le x_3, y_{31} \ge 0, y_{32} \ge 0
\end{cases}
\end{cases}
\tag{6}
$$

A run of Nash-lambda algorithm 120 generations show that (pure strategy)

$$
\boxed{
\begin{aligned}
&\left(x_1^*, x_2^*, x_3^*\right) = \left(1.0371, 0.7698, 8.1155\right) \\
&\left(y_{11}^*, y_{12}^*\right) = \left(0.6569, 0.3796\right) \\
&\left(y_{21}^*, y_{22}^*\right) = \left(0.3639, 0.3805\right) \\
&\left(y_{31}^*, y_{32}^*\right) = \left(2.1396, 5.9705\right)
\end{aligned}
}
$$

With optimal objective values

$$
\boxed{
\begin{aligned}
&y_{11}^* y_{12}^* \sin x_1 + 2y_{21}^* y_{22}^* \sin x_2 + 3y_{31}^* y_{32}^* \sin x_3 = 37. \\
&y_{11} \sin y_{12} + y_{12} \sin y_{11} = 0.4752 \\
&y_{21} \sin y_{22} + y_{22} \sin y_{21} = 0.2705 \\
&y_{31} \sin y_{32} + y_{32} \sin y_{31} = 4.3722
\end{aligned}
}
$$

The pure strategy made leader objective value reaches maximum.

A run of Nash-lambda algorithm 26 generations show that (mixed strategy)

$$\left(x_1^*, x_2^*, x_3^*\right) = (0.4000, 1.6000, 8.0000)$$
$$\left(y_{11}^*, y_{12}^*\right) = (0,0)$$
$$\left(y_{21}^*, y_{22}^*\right) = (0.7200, 0.8000)$$
$$\left(y_{31}^*, y_{32}^*\right) = (1.9968, 6.0000)$$

With optimal objective values

$$y_{11}^* y_{12}^* \sin x_1 + 2y_{21}^* y_{22}^* \sin x_2 + 3y_{31}^* y_{32}^* \sin x_3 = 36.71$$
$$y_{11} \sin y_{12} + y_{12} \sin y_{11} = 0$$
$$y_{21} \sin y_{22} + y_{22} \sin y_{21} = 1.0440$$
$$y_{31} \sin y_{32} + y_{32} \sin y_{31} = 4.9058$$

A run of Nash-lambda algorithm 44 generations show that (pure strategy)

$$\left(x_1^*, x_2^*, x_3^*\right) = (3.5833, 3.1968, 3.1999)$$
$$\left(y_{11}^*, y_{12}^*\right) = (1.9833, 1.6000)$$
$$\left(y_{21}^*, y_{22}^*\right) = (1.5968, 1.6000)$$
$$\left(y_{31}^*, y_{32}^*\right) = (1.5999, 1.6000)$$

With optimal objective values

$$y_{11}^* y_{12}^* \sin x_1 + 2y_{21}^* y_{22}^* \sin x_2 + 3y_{31}^* y_{32}^* \sin x_3 = -2.0859$$
$$y_{11} \sin y_{12} + y_{12} \sin y_{11} = 3.4482$$
$$y_{21} \sin y_{22} + y_{22} \sin y_{21} = 3.1955$$
$$y_{31} \sin y_{32} + y_{32} \sin y_{31} = 3.1985$$

The pure strategy made followers objective value reaches maximum.

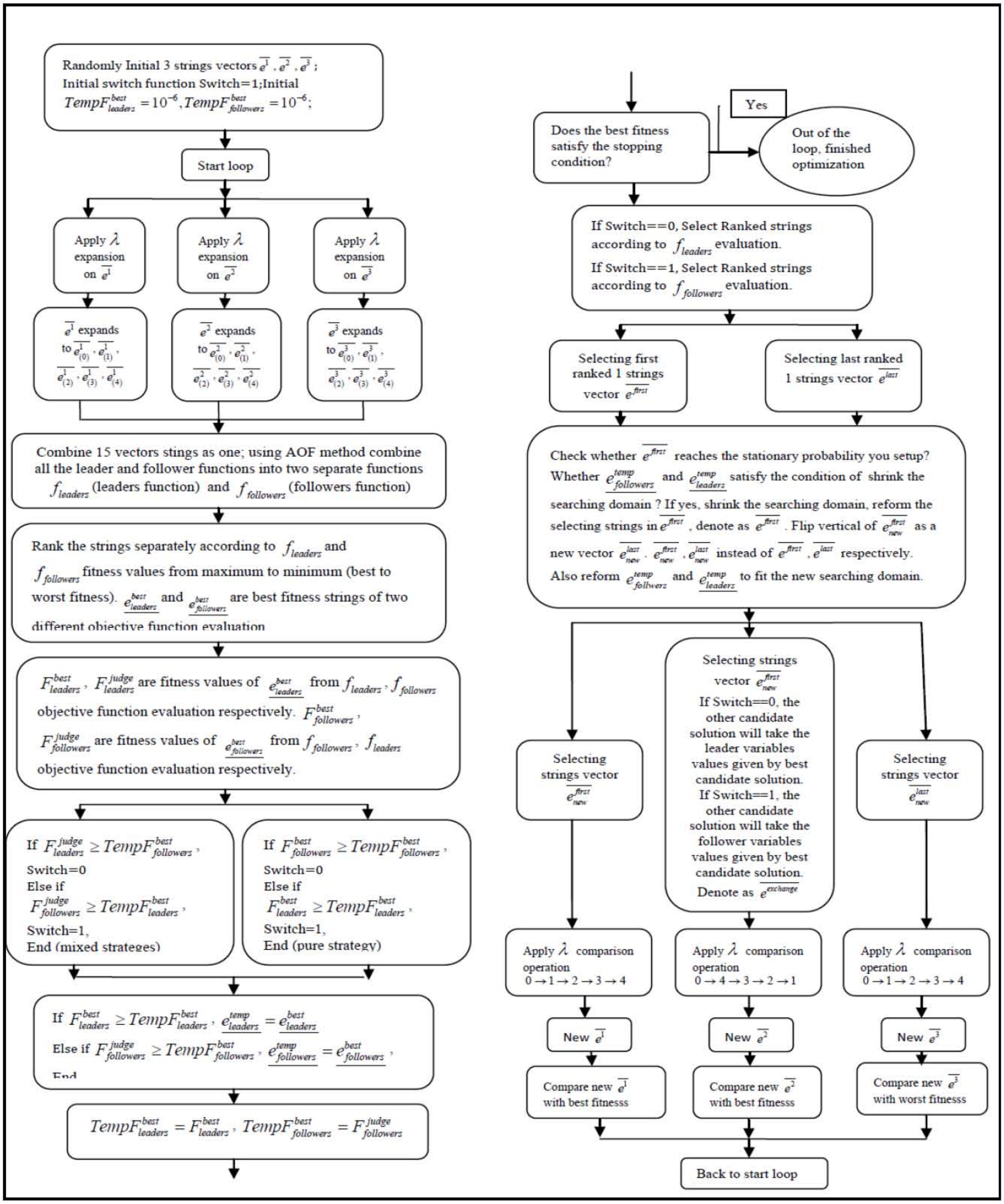



Figure 3.3.1 Bi-level programming using Nash-lambda algorithm operation process

## 3.3.5 Applications in safety and reliability

In this section, we will consider a few Nash-lambda algorithm applications in the safety and reliability field with the focus in Subsection 3.3.5.1.

### 3.3.5.1 Maintenance schedule problem

In this subsection, let us examine a maintenance scheduling application (Manbachi et al. 2010), where the authors defined a new index, named lost opportunity cost of market participation (LOCMP) since every individual generation company (GENCO) targets to maximize its profits except the reliability concerns, which is monitoring constantly by the Independent System Operator (ISO). "ISO as a market supervisor is responsible for power system reliability preservation" (Kjrulff, 1992), and therefore a player in the dynamic game of GENCO against ISO.

The strategy of each GENCO will maximize the profits at the same time will minimize the LOCMP. The LOCMP is calculated by

$$\mathrm{LOCMP} = \sum_{t}^{week} \sum_{g}^{G} \left( p_t - \left( 2\alpha\, p_{\max,g,t} + \beta \right) \right) p_{\max,g,t} h_t Y_{g,t} \tag{7}$$

where

| | |
|---|---|
| $p_t$ | Price for a strategy at time $t$ |
| $p_{\max,g,t}$ | Power generated by units in stage t (MW) |
| $\alpha, \beta$ | Cost factors (i.e., $x_1, x_2$) |
| $h_t$ | Maintenance hours of unit at stage $t$ |
| $Y_{g,t}$ | Maintenance status of units in stage $t$ (1, or 0) |

Let

| | |
|---|---|
| $C_{g,t}$ | Production cost of generation units in stage t (i.e., $x_3$) |
| $p_t^M$ | Maintenance cost of generation units |

Then the objective function for a GENCO

$$\Lambda=\sum_{t}^{week}\sum_{g}^{G}\left(\left(p_t-C_{g,t}\right)p_{\max,g,t}\left(1-Y_{g,t}\right)-p_t^{M}Y_{g,t}\right) \tag{8}$$

On the other hands, ISO as a player offers a disincentive strategy

$$p_t^{penalty}=\frac{S_t}{\sum_{t=1}^{52}S_t}C_t^{ISO\text{-}PAYMENT} \tag{9}$$

where

| | |
|---|---|
| $p_t^{penalty}$ | Penalty Index |
| $S_t$ | Quadratic Penalty Index $\left(EIR_t^{base}-EIR_t^{offered}\right)^2$ |
| $EIR_t^{base}$ | Energy Index Reliability calculated by ISO shows desirable reliability |
| $EIR_t^{offered}$ | Energy Index Reliability calculated by ISO considering offers of GENCOs |
| $C_t^{ISO\text{-}PAYMENT}$ | $\begin{cases}\text{Cost paid by ISO for penalty}\\ \text{Cost of energy not supplied}\end{cases}$ |

Then the objective function is

$$\Lambda=\sum_{t}^{week}\sum_{g}^{G}\left(\left(p_t-C_{g,t}\right)p_{\max,g,t}\left(1-Y_{g,t}\right)-p_t^{M}Y_{g,t}-p_t^{penalty}\right) \tag{10}$$

which is again a bi-level program suitable for Nash-lambda algorithm because the penalty paid by ISO needs to be minimized.

In Manbachi et al., 2010, the authors engaged a simulation approach for seeking the optimal solution. We engage the Nash-lambda scheme for searching the optimal solution. The objective function we used is

$$\Lambda=\sum_{t}^{week}\sum_{g}^{G}\left(\left(p_t-x_1p_{\max,g,t}\right)p_{\max,g,t}\left(1-Y_{g,t}\right)-x_3p_tY_{g,t}\right) \tag{11}$$

and the constraint sub-objective function is

$$\text{LOCMP}=\sum_{t}^{week}\sum_{g}^{G}\left(p_t-\left(2x_2p_{\max,g,t}+x_3\right)\right)p_{\max,g,t}h_{g,t}Y_{g,t} \tag{12}$$

and thus the bi-level program formation is

$$\begin{aligned}&\max_{x_1,x_2,x_3} \Lambda(x_1,x_2,x_3)\\&\text{s.t.}\\&\min_{x_1,x_2,x_3} \text{LOCMP}(x_1,x_2,x_3)\end{aligned} \tag{13}$$

Because we feel short of information, in the problem formulation we identify three cost variable, $x_1, x_2, x_3$ The Nash-lambda uses 36.1881seconds, 100 loops for locating the equilibrium numerical solution:

$$x_1 = 99.840, x_2 = 4.992, x_3 = 0.000 \tag{14}$$

which gives the $\max \Lambda(x_1,x_2,x_3) = 3.3816\text{E}+009$, subject to $\min \text{LOCMP}(x_1,x_2,x_3) = 8.8654\text{E}+004.$



### 3.3.5.2 Anti-terrorism

International terrorism has been a principal concern of policy makers and the public since the September 11 attack, 2001 (Heymann, 2003; Wikipedia, 2011). "The West" and the " International Terrorist Organization (ITO)" are two players in an incentive Stackelberg game model (Keohane and Zeckhauser, 2003). The objective function is

$$\Lambda(x,w,v) = -\int_0^T e^{-rt}\left(\gamma_1 x_t + \gamma_2 w_t + \gamma_3 v_t\right)dt + e^{-rT} s x_T \tag{15}$$

And thus the optimization problem is

$$\max_{v_t} \Lambda(x,w,v) \tag{16}$$

Subject to

$$\begin{aligned}&\dot{x}_t = f(x_t) - \mu w_t - g(v_t)w_t - \phi v_t + h(v_t);\\&f(x_t) = \gamma(1-x_t)x_t;\\&g(v_t) = \beta v_t;\\&h(v_t) = \alpha v_t^2,\end{aligned} \tag{17}$$

where

| | |
|---|---|
| $x_t \geq 0$ | number of terrorists at time $t$ |
| $v_t \geq 0$ | intensity of the West's terror control activities at time $t$ |
| $w_t \geq 0$ | number of ITO attacks at time $t$ |
| $f(x_t)$ | endogenous growth of ITO at time $t$ |

| | |
|---|---|
| $\mu \geq 0$ | average number of terrorists killed or arrested per attack |
| $g(v_t)$ | number of terrorists lost per terror attack due to terror control efforts $v(t)$, |
| $\phi \geq 0$ | rate at which terror control operations would deplete ITO if the West is on full counter-offensive |
| $h(v_t)$ | growth of ITO at time t due to hatred caused by collateral damage induced by (low-specificity) terror control activities of the West. |

with $\gamma$, $\beta$, and $\alpha$ being positive constants. The constraint should be the ITO wants to maximize the attacks' damages. It can be solved by a bi-level program and hence Nash-lambda algorithm is able to search its solution by changing the equality constraints into a set of inequality constraints in terms of additional explanatory variables.

### 3.3.5.3 Reliability and free riding

Another interesting of application is the problem of the reliability of public systems. It is well-known fact that the public systems cost the tax payers dearly, however, certain corner of the society (typically those never paid one cent for tax) always steal or damage these goods for self-benefiting. The problem is again a *n*-player non corporative game. Let

| | |
|---|---|
| $x_i$ | The effort tried by agent $i = 1,2;$ |
| $p(F(x_1, x_2))$ | The probability of successful operation of the system; |
| $v_i$ | The reward received by agent $i$ from successful operation of the system; |
| $c_i x_i$ | The cost paid by agent $i$ from successful operation of the system. |

Then the expected social payoff

$$p(F(x_1, x_2))(v_1 + v_2) - (c_1 x_1 + c_2 x_2) \tag{18}$$

As the specification of $F(x_1, x_2)$, Sandler and Hartley (2001) and Varian (1994) considers three regimes:

| | |
|---|---|
| Total effort: | $F(x_1, x_2) = x_1 + x_2$; |
| Weakest link: | $F(x_1, x_2) = x_1 \wedge x_2$; |
| Best shot: | $F(x_1, x_2) = x_1 \vee x_2$; |

Then the aim is

$$\max_{x_1,x_2}\left(p\left(F\left(x_1,x_2\right)\right)\left(v_1+v_2\right)-\left(c_1x_1+c_2x_2\right)\right) \tag{19}$$

The constraint is to minimize the agents' cost. The Nash equilibrium solution depends upon the regime committed. Free-riding occurs under certain conditions. However, the functional form of $F(x_1,x_2)$ in Sandler and Hartley (2001) and Varian (1994) is oversimplified, if $F(x_1,x_2)$ is non-linear in $x_1$ and $x_2$, then the Nash-lambda algorithm needs to step in for searching numerical solutions.

#### 3.3.5.4 Optimal maintenance service

In this subsection, we consider the problem of equipment maintenance by an external subcontractor. The owner of the equipment and the subcontractor are two players under non-corporative game. Both parts want the maximized profits (Jackson and Pascual, 2008). This is a bi-level program and it is appropriate to use the Nash-lambda algorithm to search optimal solution.

### 3.3.6 Conclusion

In this paper, we investigate the merging with Nash equilibrium solution with lambda algorithm, which is a type of Bayesian network (Ben-Gal, 2007; Buntine, 1994; Wikipedia, 2011; Kjrulff, 1992). We have successfully created a merged algorithm and coded it in details, i.e. at bi-level program with two players. Frankly, the numerical example in Section 3.3.4 does not link to safety and reliability. However, just this example triggered our interest to investigate the merging and programming the new algorithm because the problem formulation is strictly revealing the requirements in Definition 1. To cope with the spirit of the conference, we give a detailed reliability example in Subsection 3.3.5.1 for illustrations. In the future, we will strive to increase the number of the players first and then the 3-level, and so on. The application examples in this paper are not detailed because of the page limitation and time-constraints. We will improve the paper in this aspect.

# 3.4 Lambda Algorithm and Maximum Likelihood Estimation



Yanhong Cui, Renkuan Guo

Department of Statistical Sciences, University of Cape Town

Private Bag, Rhodes' Gift, Rondebosch 7701, Cape Town, South Africa

hadisna@hotmail.com renkuan.guo@uct.ac.za

This paper has been published in Summer Safety and Reliability Seminars, SSARS 2011, Journal of Polish Safety and Reliability Association, Vol. 1, pp. 59-72. 2010. ISBN: 978-83-925436-2-6. Editors in Chief: Krzysztof Kolowrocki, Joanna Soszynska and Enrico Zio.

### 3.4.1 Abstract

In this paper, a new global optimization algorithm by imitating ancient Chinese human body system model, named as lambda algorithm, is introduced. The lambda algorithm utilizes five-element multi-segment string to represent the n-dimensional Euclidean point and hence the string based operation rules for expansion, comparison and sorting candidate strings. The algorithm enjoys the simplest mathematical operations but generates highest searching speed and accuracy. We furthermore explore to merge the lambda algorithm with maximum likelihood procedure for creating a non-derivative scheme - likelihood- lambda procedure. An illustrative example is given.

**Keywords:** lambda algorithm, genetic algorithm, reliability, repairable system, likelihood-lambda procedure, reliability

### 3.4.2 Introduction

Safety and reliability optimization problems are fundamental components intrinsically in the sense that the statistical theory underlying them is built up by a pile of relevant mathematical optimal theories and methodologies. Particularly, in safety and reliability modelling practices the maximum likelihood estimation plays an important role. Therefore, it is necessary to improve the efficiency in searching the optimal solution of a likelihood function.

Different searching schemes have different efficiency. The standard derivative-orient scheme, the Newton-Raphson procedure is the commonly engaged, see Ben-Gal (2007), Lawless (1982), Ushakov (1994). However, more and more searching schemes are utilizing non derivative-orient schemes, for example, merging likelihood and genetic algorithm to avoid derivative computations.

The lambda algorithm is created by imitating an ancient human body system, see Cui et al. (2010), Guo et al. (2010), Kjrulff (1992). In its searching scheme, except the necessary mathematical computations for evaluating the objective function and the creation of the initial "searching population" randomly, the algorithm only involves if-else logical operation and sort procedure. In contrast to existing global optimization algorithms, particularly GA, the lambda algorithm engages the simplest mathematics but reaches the highest searching efficiency.

The remaining structure of the paper is stated as following: Section 3.4.3 serves the introduction of 5-element string and the presentation of an Euclidean vector; Section 3.4.4 will features of the lambda algorithm; Section 3.4.5 will discuss the operators engaged in the lambda algorithm and identify the Markovian features of the searching scheme; Section 3.4.6 reserves for testing the new algorithm, Section 3.4.7 proposes the merging of lambda algorithm and likelihood searching procedure and a reliability application and Section 3.4.8 concludes this paper.

### 3.4.3 Five-element string and Euclidean vector presentation

In this section we utilize the Rosenbrock's function as an example for introducing the lambda algorithm.

Assume that the two-dimensional Rosenbrock's function, $f(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + (1-x_1)^2$, which is the objective function under investigation. There are many local optimal solutions, but we are interested in obtain the global optimal solution of $f$ .

Similar to the genetic algorithm (GA) binary string representation, whose element set is $\Theta=\{0,1\}$, the λ-algorithm takes elements from a 5 element-set, which takes a membership set $\Theta=\{0, 1, 2, 3, 4\}$, to construct a string candidate solution which represents the candidate solution, Euclidean vector, $\underline{x}$, in the 2-dimensional Euclidean space $\mathbb{R}^2$ in term of a linear transformation.

**Definition 1:** A string in an algorithm, denoted by $\underline{e} = e_1 e_2 \cdots e_l$, is a sequence of $l$ elements from the membership set Θ. The total number of the elements, $l$, composed of the string $\underline{e}$ is called the length of the string.

**Definition 2:** In a string algorithm, in order to represent an n-dimensional Euclidean point $\underline{x}' = (x_1, x_2, \cdots, x_n)$, a string $\underline{e}$ is typically constituted by n segmental strings, whose length are u, i.e., the string $\underline{e} = \underline{e}_1 \underline{e}_2 \cdots \underline{e}_n$ with length $l = nu$, where the $i^{th}$ segment of the string, or the $i^{th}$ segmental string, $\underline{e}_i = e_{i,u(i-1)+1} e_{i,u(i-1)+2} \cdots e_{i,iu}$, is of length $u$, (i=1,2,$\cdots$ ,n).

**Definition 3:** A triple $(m,u,n)$ is called string configuration, where $m =$ total number of elements in the membership set Θ, $u =$ the length of segmental string $\underline{e}_i = e_{i,u(i-1)+1} e_{i,u(i-1)+2} \cdots e_{i,iu}$, and $n =$ total number of segmental strings composing of the string $\underline{e} = \underline{e}_1 \underline{e}_2 \cdots \underline{e}_n$. Let $\mathbb{S} = \{\underline{e} : e_i \in \Theta\}$ denote the string space generated from m-element membership set $\Theta_m$ and $\mathbb{S}_{(m,u,n)} = \left\{\underline{e}_1 \underline{e}_2 \cdots \underline{e}_n : e_{(i,j_i)} \in \Theta_m\right\}$ the $(m,u,n)$ configuration string space on $\Theta_m$.

**Example 4:** A string with configuration triple (5, 6, 2) $\underline{e} = \underline{e}_1\underline{e}_2$ represents a 2-dimensional candidate solution ( $X_1$, $X_2$ ) for Rosenbrock's function with two segmental strings, $\underline{e}_1$ and $\underline{e}_2$ . For example,

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 1 | 4 | 0 | 0 | 1 | 2 | 2 | 3 | 0 |

The first segmental string, $\underline{e}_1$ , is constituted by the first 6 elements, for example, $\underline{e}_1 = 1\ 3\ 4\ 1\ 4\ 0$, which represents the variable $X_1$ , while the second segmental string, $\underline{e}_2$ , is constituted by the seventh element to the twelfth element, i.e., the second six elements in the string, for example, $\underline{e}_2$=0 1 2 2 3 0, which represents the variable $X_2$. The string length $l = 2\times 6 = 12$. The correspondence between each segment string and X can be also labelled as following:

$$\underbrace{1\ 3\ 4\ 1\ 4\ 0}_{X_1}\quad \underbrace{0\ 1\ 2\ 2\ 3\ 0}_{X_2}$$

Let

| |
|---|
| $D_1 \equiv \left[u_{\min}^1, u_{\max}^1\right]$ be the searching domain of $X_1$ ; |
| $D_2 \equiv \left[u_{\min}^2, u_{\max}^2\right]$ be the searching domain of $X_2$ ; |
| $u_{\min}^1, u_{\max}^1$ be the lower bound and upper bound of $X_1$ ; |
| $u_{\min}^2, u_{\max}^2$ be the lower bound and upper bound of $X_2$ . |

Then, Eq. (1) and (2) specify the relationship between $(X_1, X_2)$ and string $\underline{e} = \underline{e}_1\underline{e}_2 = e_1e_2\cdots e_6e_7e_8\cdots e_{12}$ :

$$X_1 = u_{\min}^1 + \left(u_{\max}^1 - u_{\min}^1\right)\sum_{j=1}^{6} e_j \frac{5^{6-j}}{5^6} \tag{1}$$

and

$$X_2 = u_{\min}^2 + \left(u_{\max}^2 - u_{\min}^2\right)\sum_{j=7}^{12} e_j \frac{5^{12-j}}{5^{12}} \tag{2}$$

where $e_j,\ j = 1, 2, \cdots, 12$ denote the elements taking numbers 0, 1, 2, 3, 4 in the string.

**Definition 5:** A configuring $(i, j_i)$ indicates the $i^{th}$ segmental string and the $j^{th}$ element (position) in the $i^{th}$ segmental string.

Once we setup values of $D_1 \equiv \left[u^1_{\min}, u^1_{\max}\right]$ and $D_2 \equiv \left[u^2_{\min}, u^2_{\max}\right]$, the fitness value of a objective function $f(\underline{x})$ fitness value is readily to calculate according to Eq. (1) and (2).

A natural question will be raised: why does GA not offer the convergent power as high as the lambda algorithm does? In traditional binary strings represented algorithms, the chance of appearance of 0 or 1 element in the strings is 50% to 50% of each. The element repeated and unrepeated chances are equal, which let us feel difficult to draw any useful information from both repeated and unrepeated events respectively.

In the Table 3.4.1, we use configuration triple (2, 6, 2) to specify the string. In other words, (2, 6, 2) represents membership set Θ= {0, 1}, segmental string length 6, and 2 segmental strings.

Table 3.4.1 The binary element strings (2, 6, 2)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

Similarly, in Table 3.4.2, (5, 6, 2) represents membership set Θ={0,1,2,3,4}, unit string length 6, and 2-segmental string.

Table 3.4.2 The five-element (5,6,2) strings

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 4 | 4 | 2 | 1 | 3 | 3 | 2 | 1 | 0 | 0 |
| 2 | 1 | 1 | 3 | 1 | 0 | 4 | 3 | 2 | 3 | 0 | 1 |
| 2 | 2 | 4 | 3 | 0 | 1 | 2 | 4 | 3 | 2 | 1 | 1 |
| 1 | 2 | 4 | 3 | 0 | 1 | 2 | 4 | 3 | 2 | 1 | 1 |
| 0 | 1 | 2 | 3 | 4 | 1 | 0 | 2 | 2 | 3 | 3 | 0 |

Assume that a column of strings is ranked according to their fitness values (from minimum to maximum), if we select top 5 ranked strings as a sample, see Table 3.4.2, the first knowledge we learn from it is that their fitness values are less than any other strings. On the other hand, in the lambda algorithm, the repeated chance of each element is 20%, see Table 3.4.2. That means that in the sample, if 3 out of 5 are repeated with a particular element at any position, then the repeated chance of the element in this position is 60%, which is much higher than 20% (in binary string cases). This phenomenon is the second knowledge we would like to know. Even we might interpret this phenomenon as a consequence of randomization. However, under a perfect circulation, we might consider the extra 40% chance would be induced by their smaller fitness values. This phenomenon shows us the convergent tendency toward the optimal solution is higher than that of GA.

To highlight the global convergence tendency role in element numeration in the lambda algorithm, we divide element events into 3 categories: repeated one time element events, repeated two times element events and unrepeated element events. The lambda algorithm draws useful information from all three categories, to construct an intrinsic scheme towards global optimization.

### 3.4.4 Features of string representation

We should be aware that in searching and selecting candidate solution the λ-algorithm utilizes the information contained in the value of an element, the element position in the segmental string, and the sequential order of the segmental string. Note that the transformation matrix plays a vital role for linking a string of length $nu$, $\underline{e} = \underline{e}_1\underline{e}_2\cdots\underline{e}_n$, where the $i^{th}$ piece of the string, or the $i^{th}$ segmental string, with length $u$, $\underline{e}_i = e_{i,u(i-1)+1}e_{i,u(i-1)+2}\cdots e_{i,iu}$, (i=1,2,$\cdots$ ,n), and an n-dimensional Euclidean point $\underline{x}' = (x_1, x_2, \cdots, x_n)$. In other words, the global search strength mechanism lies on that the weighting system, i.e., $\left\{\frac{5^5}{5^6},\frac{5^4}{5^6},\frac{5^3}{5^6},\frac{5^2}{5^6},\frac{5^1}{5^6},\frac{5^0}{5^6},0,0,0,0,0,0\right\}$, assigned to the 6 members in the first segmental string and $\left\{0,0,0,0,0,0,\frac{5^5}{5^6},\frac{5^4}{5^6},\frac{5^3}{5^6},\frac{5^2}{5^6},\frac{5^1}{5^6},\frac{5^0}{5^6}\right\}$, the weighting system assigned to the 6 members in the second segmental string create the possibility that change in the element $j_i$ (position) of the $i^{th}$ segmental string will have different impacts because different position has different weight. A $(5,6,2)$ string, denoted by $e_1e_2\cdots e_6e_7,e_8\cdots e_{12}$, the blue-color elements are the first segmental string, representing $x_1$, the red-color elements are the second segmental string, representing $x_2$.

Logically, changes in elements $e_1$ and $e_7$ will result in large changes in $x_1$ and $x_2$ respectively, because the highest weight 0.2 is assigned to them, while changes in $e_6$ and $e_{12}$ will result in the smallest changes in $x_1$ and $x_2$ respectively, because the lowest weight 0.000064 is assigned to them. Therefore, a well-constructed string element shift scheme will have a balanced global searching capability as well as local fine-tune capacity.

**Example 6:** Define $u_{\min} = -10^{10}$, $u_{\max} = +10^{10}$, then $u_r = u_{\max} - u_{\min} = 2\times10^{10}$. String 1 in Table 3: 1 2 4 3 0 1 2 4 3 2 11 is the base for observing the impacts from string element changes. String 2 changes the first element of the first segmental string in String 1 by adding 1 and the first element of the second segmental string by adding 1, which is the smallest shift in size at highest weight 0.2. The change in $x_1$ and $x_2$ is quite large with distance 5656854249.5. However, String 3 changes the sixth element of the first segmental string by adding 3 and the sixth element of the second segmental string by adding 3, which is the largest shift in size at highest weight 0.000064. The change in $x_1$ and $x_2$ is much small with distance 202276452.4. Table 3.4.3 lists the changing effects for comparisons.

Table 3.4.3 The impacts of weights in global searching and local turn-up

| String | $x_1$ | $x_2$ | $\lvert\Delta x\rvert$ |
|---|---|---|---|
| 1 2 4 3 0 1 2 4 3 2 1 1 | -3662720000 | 1751680000 | |
| 2 2 4 3 0 1 3 4 3 2 1 1 | 337280000 | 5751680000 | 5656854249.5 |
| 1 2 4 3 0 4 2 4 3 2 1 4 | -3460480000 | 1755520000 | 202276452.4 |

It is critical to emphasize here that the global feature of a string algorithm is also materialized by the range parameter setting, i.e., the setting of $u_{\min}, u_{\max}$ . The larger $u_r = u_{\max} - u_{\min}$, the better global coverage is intended.

Examining the $i^{th}$ component of $\underline{x}$, $x_i$, which is a real number, for example, $x_i = 7.5673$, traditional optimization schemes may be shift $x_i = 7.5673$ by a small increment, say, 0.005, it is intuitive that by changing at digit level of a real, the scheme remains at a local algorithm nature. Therefore, the globally and locally simultaneous searching feature of the string scheme will definitely speed up the global optimization algorithm. By repeated and unrepeated (string) elements numerated from high weight position (i.e. element of a segmental string) to lower weight position within the strings (the candidate solution strings), in terms of sort and pack scheme repeatedly until the final optimized solution is found. Our computation experiences show that in searching the final global optimal solution, it is often the case that the first 3 or 4 positions (high weight elements) in a string (if the string size is 6) could adequately secure the main searching path toward the final optimal solution globally. Nevertheless, the lower cell elements are able to change for fine-tuning $x_i$ in small increment manner, which allows the searching scheme to access the final optimal solution with amazingly high precision.

### 3.4.5 Operators in lambda algorithm

In this section we will systematically explore the mathematical operators in the lambda algorithm. For this purpose, it is necessary to introduce more notations.

#### 3.4.5.1 String vector

A string vector, denoted by $\overline{e^1}$, is a column vector taking strings as its components. The dimensionality indicates how many strings are used to construct a string vector. It is obvious that for (5, 6, 2) configuration string vector, $\overline{e^1}$ of dimensionality 100, it represents as 100 pairs of variables $X_1$ and $X_2$. As matter of fact, a string vector $\overline{e^1} = \left(e_{ij}\right)_{N\times l}$ is a matrix of elements $e_{ij} \in \Theta_5$ with size $N \times l$. *Figure 1* intuitively a string vector of $\left(e_{ij}\right)_{100\times 12}$:

$$\overline{e^1} = \begin{bmatrix} \underline{e}_1^1 \\ \underline{e}_2^1 \\ \vdots \\ \underline{e}_{100}^1 \end{bmatrix} = \begin{bmatrix} e_{1,1} & \cdots & e_{1,6} & e_{1,7} & \cdots & e_{1,12} \\ e_{2,1} & \cdots & e_{2,6} & e_{2,7} & \cdots & e_{2,12} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ e_{100,1} & \cdots & e_{100,6} & e_{11,7} & \cdots & e_{100,12} \end{bmatrix} \tag{3}$$



#### 3.4.5.2 λ operator and $\lambda^{-1}$operator

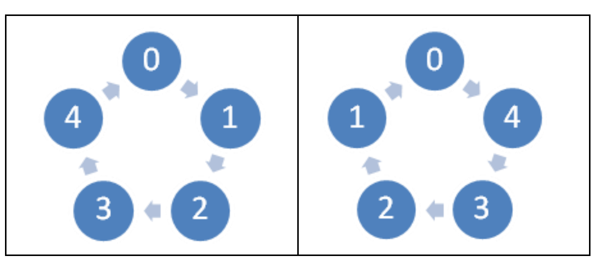



Figure 3.4.1 Cycle behaviour of operator $\lambda[e_i]$ (Left) and operator $\lambda^{-1}[e_i]$ (Right)

**Definition 7:** λ operator for (5, u, n) configuration string element is defined by

$$\lambda[e_i] = \begin{cases} e_i + 1 \; if \;\; e_i \in \{0,1,2,3\} \\ 0 \qquad\quad if \;\; e_i \in \{4\} \end{cases} \tag{4}$$

where $e_i$ is an element in a string $\underline{e}$.

**Definition 8:** $\lambda^{-1}$ operator for (5, u, n) configuration string element is defined by

$$\lambda^{-1}[e_i] = \begin{cases} e_i - 1 \; if \;\; e_i \in \{1,2,3,4\} \\ 4 \qquad\quad if \;\; e_i \in \{0\} \end{cases} \tag{5}$$

where $e_i$ is an element in a string $\underline{e}$.

### 3.4.5.3 Cyclic behaviour of a vector string

**Definition 9:** Let $\underline{e} = e_1 e_2 \cdots e_l$ be a (5, u, n) configuration string, the λ operation on string $\underline{e}$ is

$$\lambda[\underline{e}] \triangleq \lambda[e_1]\lambda[e_2]\cdots\lambda[e_l]. \tag{6}$$

**Theorem 10:** Let $\underline{e} = e_1 e_2 \cdots e_l$. be a $(5, u, l/u)$ configuration string of length $l = nu$, where u is the length of segmental string. Then

$$\lambda^{(n)}[\underline{e}] = \lambda^{(\mathrm{mod}_4(n))}[\underline{e}], \tag{7}$$

where

$$\begin{aligned} \lambda^{(n)}[\underline{e}] &\triangleq \underbrace{\lambda\left[\lambda\left[\cdots\lambda[\underline{e}]\right]\right]}_{\text{repeat } n \text{ times}}, \\ \lambda^{(0)}[\underline{e}] &\equiv \underline{e}. \end{aligned} \tag{8}$$

(Note that $\mathrm{mod}_4(\cdot)$ is a modulo operator using 4 as its quotient.)

**Proof:** According to Definition 7, applying λ operator to a string $\underline{e}$ is just applying the operator λ to each individual element in string $\underline{e}$, $e_i$, without any disturbance on the sequential order, $i$. Also, recall that $\lambda^{(n)}[e_i]$ is cyclic as *n* increases by step size 1, e.g., $e_i = 0$, $\lambda^{(5)}[e_i] = \lambda^{(0)}[e_i] = e_i = 0$. Hence, $\lambda^{(n)}[\underline{e}] = \lambda^{(\mathrm{mod}_4(n))}[\underline{e}]$.

**Corollary 11:** Let $\overline{e^1}$ be a string vector. Then

$$\begin{aligned} &\lambda^{(n)}\left[\overline{e^1}\right] = \lambda^{(\mathrm{mod}_4(n))}\left[\overline{e^1}\right], \\ &\lambda^{(0)}\left[\overline{e^1}\right] \equiv \overline{e^1}. \end{aligned} \tag{9}$$

**Remark 12:** Define $\overline{e^1_{(0)}} \triangleq \overline{e^1}$, $\overline{e^1_{(1)}} \triangleq \lambda\left[\overline{e^1}\right] = \lambda\left[\overline{e^1_{(0)}}\right]$, $\overline{e^1_{(2)}} = \lambda\left\lfloor \lambda\left[\overline{e^1}\right]\right\rfloor = \lambda^{(2)}\left[\overline{e^1}\right]$, $\overline{e^1_{(3)}} = \lambda^{(3)}\left[\overline{e^1}\right]$, and $\overline{e^1_{(4)}} = \lambda^{(4)}\left[\overline{e^1}\right]$. We treat the string vectors within one cycle as the 5 states of $\overline{e^1}$ : { $\overline{e^1_{(0)}}$ , $\overline{e^1_{(1)}}$ , $\overline{e^1_{(2)}}$ , $\overline{e^1_{(3)}}$ , $\overline{e^1_{(4)}}$ }, see Figure 3.4.2. Then if we prepare *M* string vectors, then in terms of λ operator, we will immediately expand to 5*M* string vectors. In such a sense, we call it as $\lambda$ expansion operation on string vectors.

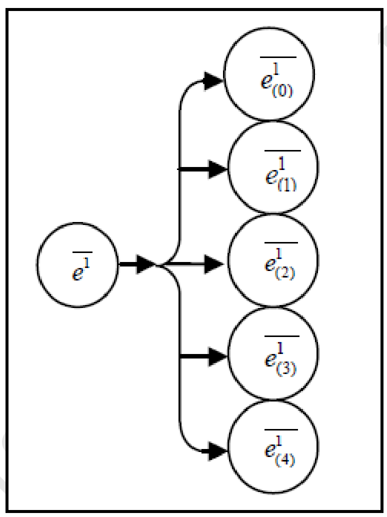


Figure 3.4.2 A string vector $\overline{e^1}$ and its cyclic vectors

**Example 13:** Given a string $\underline{e^1_1}$ in strings vector $\overline{e^1}$ :

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 1 | 4 | 0 | 0 | 1 | 2 | 2 | 3 | 0 |

The string $\underline{e^1_1}$ will be expanded by applying $\lambda$ , the cyclic vectors with respect to $\underline{e^1_1}$ are:

String $\underline{e^1_{1(0)}}$ of (0) state in the string vector $\overline{e^1_{(0)}}$ ,

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 4 | 0 | 2 | 0 | 1 | 1 | 2 | 3 | 3 | 4 | 1 |

String $\underline{e^1_{1(1)}}$ of (1) state in string vector $\overline{e^1_{(1)}}$ ,

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 0 | 1 | 3 | 1 | 2 | 2 | 3 | 4 | 4 | 0 | 2 |

String $\underline{e^1_{1(2)}}$ of (2) state in the string vector $\overline{e^1_{(2)}}$ ,

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 1 | 2 | 4 | 2 | 3 | 3 | 4 | 0 | 0 | 1 | 3 |

String $\underline{e^1_{1(3)}}$ of (3) state in the string vector $\overline{e^1_{(3)}}$,

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 0 | 3 | 4 | 4 | 0 | 1 | 1 | 2 | 4 |

All the strings generated will be in the same row position in relevant string vectors.

### 3.4.5.4 λ Comparison operation

In the candidate solution search procedure, we need to compare the objective function $f(\underline{x}), \underline{x} \in \mathbb{R}^n$ at different values of candidates $\underline{x}$ and thus it is inevitably to compare strings, at which $f$ will be optimal.

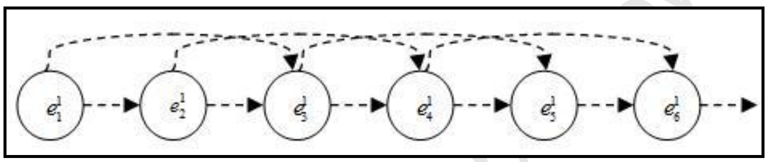



Figure 3.4.3 λ comparison operation of string vector $\overline{e^1}$

There are two kinds of λ comparison operations, but we only engage the first kind of comparison operation within a strings vector.

**Definition 14:** λ comparison operation of the first kind means that the value of the element $e_i$ follows $0 \to 1 \to 2 \to 3 \to 4$ criterion to change.

**Example 15:** Let $\underline{e^1_1}, \underline{e^1_2}, \cdots\cdots, \underline{e^1_{100}}$ are strings in string vector $\overline{e^1}$. Let $e^1_{n(1)}, e^1_{n(2)}, \cdots\cdots, e^1_{n(12)}$ are elements in any string $\underline{e^1_n}$, respectively, $n = 1, 2, \cdots, 100$. Then

For n=1:1:100-2
For i=1:1:12
If $e^1_{n(i)} \equiv e^1_{n(i+1)}$
$e^1_{n(i+1)} \leftarrow \lambda\left[e^1_{n(i)}\right]$
ELSE IF
$e^1_{n(i)} \equiv e^1_{n(i+2)}$
$e^1_{n(i+2)} \leftarrow \lambda\left[e^1_{n(i)}\right]$
END

**Definition 16:** λ comparison operation of the second kind means that the value of the element $e_i$ follows $0 \to 4 \to 3 \to 2 \to 1$ criterion to change.

**Example 17:** Let, $\underline{e}_1^1, \underline{e}_2^1, \cdots\cdots, \underline{e}_{100}^1$ are strings in string vector $\overline{e^1}$. Let $e_{n(1)}^1, e_{n(2)}^1, \cdots\cdots, e_{n(12)}^1$ are elements in any string $\underline{e}_n^1$, respectively, $n = 1, 2, \cdots, 100$. Then

For n=1:1:100-2
For i=1:1:12
If $e_{n(i)}^1 \equiv e_{n(i+1)}^1$
$e_{n(i+1)}^1 \leftarrow \lambda^{-1}\left[e_{n(i)}^1\right]$
ELSE IF
$e_{n(i)}^1 \equiv e_{n(i+2)}^1$
$e_{n(i+2)}^1 \leftarrow \lambda^{-1}\left[e_{n(i)}^1\right]$
END

Furthermore, we state the assumption on the initial set of string vectors.

Initialization Assumption**:** Let the initial string vector be $\overline{e^0} = \left(e_{ij}^{(0)}\right)_{M \times l}$ such all the elements in the $i^{th}$ string $\underline{e}_i$ are mutually independent, $i = 1, 2, \cdots, M$.

Let $D \equiv [u_{\min}, u_{\max}]^n$ be the searching domain for an objective function $f(\underline{x})$ defined in n-dimensional Euclidean space $\mathbb{R}^n$. It is obvious D determines the scope of searching globally. Mathematically, the linear system linking the strings $\underline{e}$ and the system state $\underline{x}$ can be expressed by

$$\begin{cases} x_1 = u_{\min} + (u_{\max} - u_{\min}) \sum_{j=1}^{u} e_j \dfrac{5^{u-j}}{5^u} \\ x_2 = u_{\min} + (u_{\max} - u_{\min}) \sum_{j=u+1}^{2u} e_j \dfrac{5^{2u-j}}{5^{2u}} \\ \qquad\vdots \\ x_n = u_{\min} + (u_{\max} - u_{\min}) \sum_{j=(n-1)u+1}^{nu} e_j \dfrac{5^{nu-j}}{5^{nu}} \end{cases} \tag{10}$$

Let the weight matrix be

$$O_{n\times nu}=\begin{bmatrix} \frac{5^{u-1}}{5^u} & \cdots & \frac{5^0}{5^u} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \frac{5^{u-1}}{5^u} & \cdots & \frac{5^0}{5^u} & 0 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \frac{5^{u-1}}{5^u} & \cdots & \frac{5^0}{5^u} \end{bmatrix} \tag{11}$$

And further, let

$$\underline{x}=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix};\ \underline{u}_{\min}=\begin{bmatrix} u_{\min} \\ u_{\min} \\ \vdots \\ u_{\min} \end{bmatrix};\ u_r=u_{\max}-u_{\min} \tag{12}$$

and write string $\underline{e}=e_1e_2\cdots e_l$ in column vector form, (i.e., $nu\times 1$ column vector of element $e_i$), that is

$$\underline{e}'_{nu\times 1}=\begin{bmatrix} \underline{e}'_1 \\ \vdots \\ \underline{e}'_u \\ \underline{e}'_{u+1} \\ \vdots \\ \underline{e}'_{2u} \\ \vdots \\ \underline{e}'_{nu-1} \\ \vdots \\ \underline{e}'_{nu} \end{bmatrix} \tag{13}$$

Then the candidate solution is a linear transformation of the (5, *u*, *nu*) configured string representation

$$\underline{x}=\underline{u}_{\min}+u_r O_{n\times nu}\underline{e}'_{nu} \tag{14}$$

**Definition 18:** Let $\underline{x}^{(q)}=\underline{u}_{\min}+u_r\lambda^{(q)}\left[\underline{e}\right]O', q\in\{0,1,2,3,4\}$, then,

$$\left\{ f\left(\underline{x}^{(0)}\right), f\left(\underline{x}^{(1)}\right), f\left(\underline{x}^{(2)}\right), f\left(\underline{x}^{(3)}\right), f\left(\underline{x}^{(4)}\right) \right\}_{\underline{e}} \tag{15}$$

is called the cyclic set of objective function values with respect to string $\underline{e}'_{nu\times 1}$.

**Definition 19:** ($\lambda$ comparison operation on two strings) If we compare 2 strings $\underline{e_1}$ and $\underline{e_2}$. Assume that String $\underline{e_1}$ (candidate solution)'s fitness value is better than string $\underline{e_2}$'s, and then we managed some change to $\underline{e_2}$. Let $L$ be the length of strings $\underline{e_1}$, $\underline{e_2}$. Let $e_{1i}$ and $e_{2i}$ be one of the element of $\underline{e_1}$ and $\underline{e_2}$ respectively, and $g_1(\cdot)$ and $g_2(\cdot)$ be 2 different kind of $\lambda$ comparison operation functions.

$$g(e_{2i}) = \begin{cases} \lambda[e_{2i}], & if\ e_{1i} = e_{2i} \\ e_{2i}, & if\ e_{1i} \neq e_{2i} \end{cases} \tag{16}$$

and

$$g_2(e_{2i}) = \begin{cases} \lambda^{-1}[e_{2i}], & if\ e_{1i} = e_{2i} \\ e_{2i}, & if\ e_{1i} \neq e_{2i} \end{cases} \tag{17}$$

Because $\lambda$ comparison operation have 2 different kind of functions, and each function apply on only one single string vector $\bar{e}$, so we may only describes $g(\cdot)$ „s applications as a sample for other similar function.

In Definition 19, we only describe one kind $\lambda$ comparison function $g(\cdot)$. The other one function is similar to $g(\cdot)$ operation in string vector $\bar{e}$.

**Definition 20:** Given a string vector $\underline{e}_{vu\times N}$. A string $\underline{e} = e_1 e_2 \cdots e_u e_{u+1} \cdots e_{2u} \cdots e_{(v-1)u+1} \cdots e_{vu}$ represent a candidate solution has n components. By ranking of the fitness values from best to worst, we have a sorted string vector $\overline{e'}$, where $e_{ni}$ denotes any element in $\overline{e'}$ at $n^{th}$ row, $l^{th}$ column. Then $\lambda$ Comparison operation in strings vector defined as:

If $n \geq 3$

$$g(e_{nl}) = \begin{cases} \lambda[e_{nl}], & if\ e_{nl} = e_{(n-1)l} \neq e_{(n-2)l} \\ \lambda[e_{nl}], & if\ e_{nl} \neq e_{(n-1)l} = e_{(n-2)l} \\ \lambda^{(2)}[e_{nl}], & if\ e_{nl} = e_{(n-1)l} = e_{(n-2)l} \\ e_{nl}, & if\ e_{nl} \neq e_{(n-1)l} \neq e_{(n-2)l} \end{cases} \tag{18}$$

where

$$\begin{aligned} p(\lambda[e_{nl}]) &= p\left(e_{nl} = e_{(n-1)l} \neq e_{(n-2)l}\right) + p\left(e_{nl} \neq e_{(n-1)l} = e_{(n-2)l}\right) \\ &= 0.16 + 0.16 = 0.32; \\ p\left(\lambda^{(2)}[e_{ni}]\right) &= 0.04, p(e_{ni}) = 0.64 \end{aligned} \tag{19}$$

If $n = 2$

$$g(e_{nl}) = \begin{cases} \lambda[e_{nl}], & if\ \mathrm{e}_{nl} = e_{(n-1)l} \\ e_{nl}, & if\ \mathrm{e}_{nl} \neq e_{(n-1)l} \end{cases} \tag{20}$$

where

$$p(\lambda[e_{nl}]) = 0.2 \text{ and } p(e_{nl}) = 0.8 \tag{21}$$

If $n = 1$

$$g(e_{nl}) = e_{nl} \tag{22}$$

Note that at each looping time $t$, a $\lambda$ comparison operation on whole string vector $\bar{e}$ will result in a new conditional random variable. If denote it as $\bar{e}_t$, $t = 0, 2, \cdots, n^0$, then $\{\bar{e}_t, t = 0, 2, \cdots, n^0\}$ is a stochastic process and furthermore it is a Markov Markov (decision) process due to the independent Initialization Assumption. Because the decision for choosing actions ( $\lambda$ comparison operation) does not only depend on the present state but also concerning prior states, so the process is not a simple Markov decision process, but more complicated.

### 3.4.5.5 λ Comparison operation

**Definition 21:** Given a string $\underline{e}_{nu \times 1}$ in column vector form, then the set of strings after an expansion

$$\lambda^{\text{expansion}}\left[\underline{e}^{'}\right] \triangleq \left\{\lambda^{(0)}\left[\underline{e}^{'}\right], \lambda^{(1)}\left[\underline{e}^{'}\right], \lambda^{(2)}\left[\underline{e}^{'}\right], \lambda^{(3)}\left[\underline{e}^{'}\right], \lambda^{(4)}\left[\underline{e}^{'}\right]\right\} \tag{23}$$

is called the $\lambda$ expansion set.

Now we would like to examine the string state change in $\lambda$ expansion set after a $\lambda$ comparison operation executed in string vector. If $n \geq 3$

$$\lambda^{\text{expansion}} g\left(e_{nl}\right) = \begin{cases} \lambda^{(k)}\left[\underline{e}_n\right]\lambda\left[e_{nl}\right], & if\ e_{nl} = e_{(n-1)l} \neq e_{(n-2)l} \\ \lambda^{(k)}\left[\underline{e}_n\right]\lambda\left[e_{nl}\right], & if\ e_{nl} \neq e_{(n-1)l} = e_{(n-2)l} \\ \lambda^{(k)}\left[\underline{e}_n\right]\lambda^{(2)}\left[e_{nl}\right], & if\ e_{nl} = e_{(n-1)l} = e_{(n-2)l} \\ \lambda^{(k)}\left[\underline{e}_n\right], & if\ e_{nl} \neq e_{(n-1)l} \neq e_{(n-2)l} \end{cases} \tag{24}$$

If $n = 2$

$$\lambda^{\text{expansion}}\left(g\left(e_{nl}\right)\right) = \begin{cases} \lambda^{(k)}\left[\underline{e}_{nl}\right]\lambda\left[\underline{e}_{nl}\right], & if\ \mathrm{e}_{nl} = e_{(n-1)l} \\ \lambda^{(k)}\left[\underline{e}_{nl}\right], & if\ \mathrm{e}_{nl} \neq e_{(n-1)l} \end{cases} \tag{25}$$

If $n = 1$

$$\lambda^{\text{expansion}}\left(g\left(e_{nl}\right)\right) = \lambda^{(k)}\left[\underline{e}_n\right] \tag{26}$$

where $k = 0,1,2,3,4;\ n = 1,2,\cdots N;\ l = 1,2,\cdots L$. $L$ is the length of string, $N$ represent size of strings in string vector.

Both $\lambda$ expansion and $\lambda$ comparison operations in string vector are taken after ranking the string vector according to the value of objective function $f\left(\underline{x}\right)$. After ranking, the fitness values corresponding to strings $e_n, e_{n-1}, e_{n-2}$ are supposed to be very close to those corresponding to whole vector strings.

Therefore, what we need to find out are whether or not some same elements exist in each of $\underline{e}_n, \underline{e}_{n-1}, \underline{e}_{n-2}$ (three strings) to ensure those repeated elements in the strings are the reason why the fitness values are similar. According to Eq. (20), we can see the repeated elements $e_{nl}$ already separated from the unrepeated elements, by taken an extra $\lambda\left[\cdot\right]$ operation, the twice time repeated elements also separated from the unrepeated elements by taken two times $\lambda\left[\cdot\right]$ operation. Then one time and two times repeated elements rejoin with other elements in $\lambda^{(k)}\left[\underline{e}_n\right]$, *k=0, 1,2,3,4* respectively.

Consequently, we can select only one string from rejoined 5 states $\lambda^{(k)}\left\lfloor \underline{e_n^{new}} \right\rfloor$, $k=0,1,2,3,4$ of strings. After carrying on the above process recursively, the sequence of the fitness values of objective function will be convergent. The recursive procedure is shown in Figure 3.4.4, which demonstrates a dynamic Bayesian network pattern. Figure 3.4.5 gives the flow chart to express the operations process of λ algorithm.

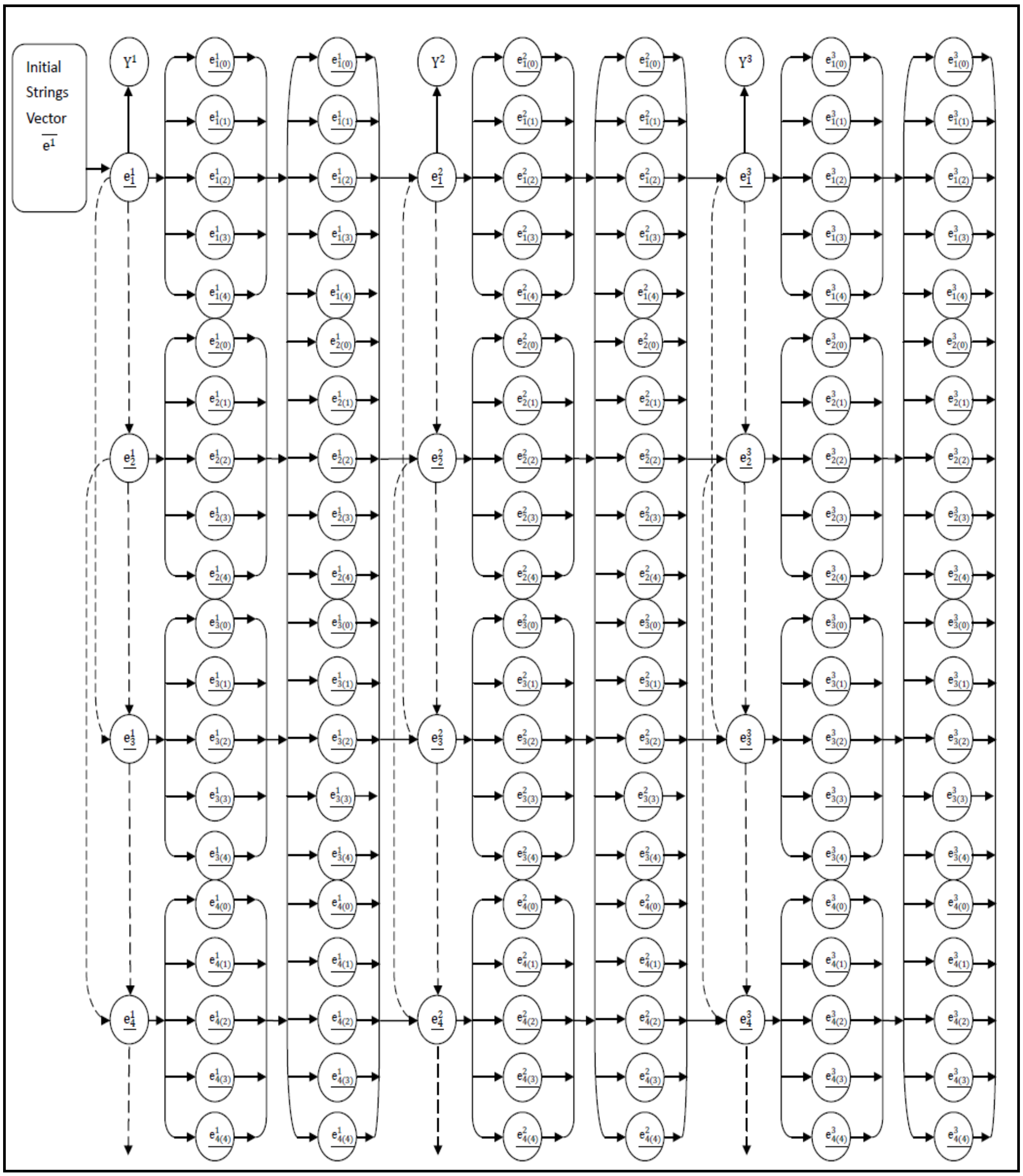



Figure 3.4.4 A dynamic Bayesian networks (DBNs) representation of λ algorithm

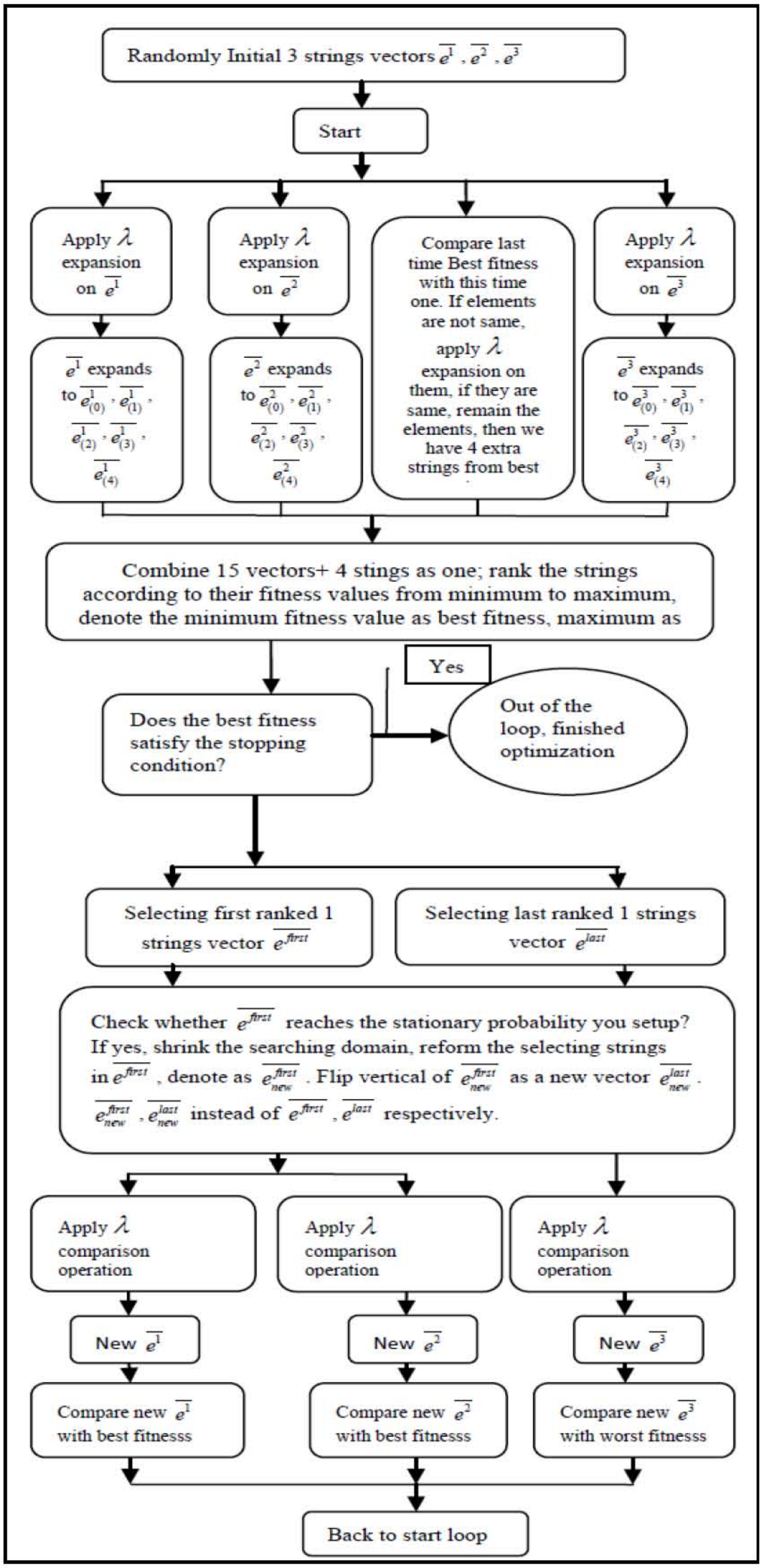



Figure 3.4.5 Flow chart to express the operations process of $\lambda$ algorithm

Bayesian networks (BNs) is a probabilistic graphical model (GM), where an individual node in the GM represents a random variable, while those edges between the nodes represent the conditional probabilities among the corresponding random variables. A GM enjoys certain degree of Markov property. See Blische and Murthy (2000), Buntine (1994), Guo et al. (2010).

A dynamic Bayesian networks (DBNs) is a general state-space model as an extension of Kalman Filter Models and Hidden Markov model. General speaking, a state-space model first specifies a prior $p(X_0)$ and a state-transition function, $p(X_t \mid X_{t-1})$, and an observation function, $p(Y_t \mid X_{t-1})$. It is critical that the observations are conditional first-order Markov $p(Y_t \mid X_t, Y_{t-1}) = p(Y_t \mid X_t)$. The Markovian character of DBNs essentially guarantees the existence of the stationary probability of the steady state.

It is fundamental to recognize that $\lambda$ algorithm engages a mechanism of the DBNs. Such recognition drove out the long-time bothering issue, why a $\lambda$ algorithm converges almost sure and the global optimization can be achieved.

### 3.4.6 Testing examples

As a conventional step to bring in a new global optimization algorithm, we utilize the new algorithm to search the optima of four 30-dimensional testing functions and three 10-dimensional test functions. In addition, we use two extreme challengeable testing functions. The string configuration for the lambda algorithm engaged for the first three testing is (5, 4,120), but for Levy function is (5, 3, 90).

Table 3.4.4 lists conventional test indices for the four 30-dimensional test functions.

Table 3.4.4 Algorithm efficiency indices I

| Search indices | Ackley | Dixon & Price | Griewank | Levy |
|---|---|---|---|---|
| Domain | $[-15,30]^{30}$ | $[-10,10]^{30}$ | $[-600,600]^{30}$ | $[-10,10]^{30}$ |
| Time (sec.) | 180.32 | 167.12 | 74.41 | 104.94 |
| Loop | 144 | 289 | 125 | 187 |
| Probab. control | 0.9 | 0.8 | 0.8 | 0.8 |

As to the function specifications and searched optima for the four 30-dimensional test functions, we list them as following:

**1. Ackley function:** Number of variables: $n$ = 30. The minimum is 0 when $x_i = 0$, $i = 1,\cdots,30$. Ackley function in general takes the form:

$$f(x_1,\cdots,x_n) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e \tag{27}$$

The searched minimum =7.9936E-015;

Optimal $\underline{x}^o$ =1.0E-014 $\times$ (0.0391 0.0857 0.1666 0.1666 0.1767 -0.1808 0.1631 0.1768 0.1790 -0.5370 0.2282 0.4305 -0.6646 0.1693 0.1778 -0.1060 0.0743 0.1775 0.0127 0.1605 -0.0093 0.1357 0.0338 0.1007 0.0285 -0.3655 -0.0357 -0.3171 -0.1896 0.0224)

**2. Dixon and Price Function:** Number of variables: $n$ = 30, the minimum is 0 when $x_i = 0$, $i = 1,\cdots,30$. The function is defined by

$$f(x_1,\cdots,x_n) = (x_1 - 1)^2 + \sum_{i=2}^{n} i\left(2x_i^2 - x_{i-1}\right)^2 \tag{28}$$

Searched minimum =0.7463, optimal solution is

$\underline{x}^o$ =(0.2399 0.0866 -0.0012 -0.0004 -0.0050 -0.0004 -0.0026 -0.0046 -0.0116 -0.0029 0.0030 0.0023 0.0055 -0.0011 -0.0014 0.0005 0.0000 0.0017 0.0225 -0.0161 0.0004 -0.0011 -0.0002 0.0289 0.0161 -0.0001 -0.0009 -0.0005 -0.0023 0.0003)



**3. Griewank Function:** Number of variables: $n$ =30. The minimum is 0 when $x_i = 0$. The $n$ -dimensional Griewank function takes the form:

$$f(x_1,\cdots,x_n) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(x_i / \sqrt{i}\right) + 1 \tag{29}$$

Searched minimum =0

Optimal $\underline{x}^o$ =1.0E-007 *(0.0007 -0.0075 -0.0229 0.0295 0.0282 0.0106 -0.0046 0.0347 -0.0426 -0.0098 -0.0229 0.0636 -0.0007 -0.0098 0.0374 -0.0033 0.0111 0.0754 -0.0033 0.0164 0.0004 0.1540 0.2458 -0.0885 0.0360 0.1475 0.0020 -0.3008 0.0492 0.0557)

**4. Levy function:** Number of variables: $n$ =30. The minimum is 0 when $x_i = 1$

$$\begin{aligned} f(x_1,\cdots,x_n) &= \sum_{i=0}^{n-2} (y_i - 1)^2 \left(1 + 10\sin^2(\pi y_i + 1)\right) \\ &+ \sin^2(\pi y_0) + + (y_{n-1} - 1)^2 \left(1 + \sin^2(2\pi x_{n-1})\right) \end{aligned} \tag{30}$$

where

$$y_i = 1 + \frac{x_i - 1}{4},\ i = 1, \cdots, n \tag{31}$$

Searched minimum =0.5840, optimal solution is

| |
|---|
| $\underline{x}^o$ = (1.0166 0.9980 0.9982 0.9919 0.9978 0.2377 0.3999 0.9944 1.0103 0.9965 0.9992 1.0105 0.9773 1.0031 0.3994 1.0407 1.0140 -0.0792 1.0009 0.9985 0.9957 1.0060 1.0487 0.9937 0.3999 1.0037 0.9966 0.3933 0.3999 1.0108) |

Table 3.4.5 summarizes three 10-dimensional test functions. The search scheme utilizes (6, 4, 40) string configuration.

Table 3.4.5 Algorithm efficiency indices II

| Search indices | Michalewics | Rastrigin | Rosenbrock |
|---|---|---|---|
| Domain | $[0, \pi]^{10}$ | $[-5, 5]^{10}$ | $[-5, 5]^{10}$ |
| Time (sec.) | 90.66 | 37.82 | 24.47 |
| Loop | 200 | 174 | 100 |
| Probability control | 0.8 | 0.4 | 0.98 |



**5. Michalewics Function:** Number of Variables: $n$ = 10. The theoretical minimum value is -9.66015

$$f(x_1, \cdots, x_n) = -\sum_{i=1}^{n} \sin(x_i)\left(\sin\left(ix_i^2 / \pi\right)\right)^{20} \tag{32}$$

The searched minimum value = - 9.2562, the optimal value is

| |
|---|
| $\underline{x}^o$ = (2.1987 1.5692 2.2179 1.9225 0.9947 1.5733 1.4516 1.7603 1.6588 1.2171) |

**6. Rastrigin Function:** Number of Variables: $n$ =10. The theoretical minimum value is 0 when $x_i = 0$, $i = 1, \cdots, 10$.

$$f(x_1, \cdots, x_n) = -10n + \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i)\right) \tag{33}$$

The searched minimum =0, the optimal solution is

| |
|---|
| $\underline{x}^o$ =1.0E-008 × (0.4096 -0.0819 -0.0819 0.2458 0.2458 -0.2130 0.1147 -0.1802 -0.4096 0.2458) |

**7. Rosenbrock Function:** Number of Variables: $n$ = 10. The general form of Rosenbrock Function is

$$f(x_1,\cdots,x_n)=\sum_{i=1}^{n-1}\left[100\left(x_i^2-x_{i+1}\right)^2+\left(x_i-1\right)^2\right] \tag{34}$$

The theoretical minimum is 0, when $x_i = 1$, $i = 1,\cdots,10$. The searched minimum = 0.000194808, the optimal

| |
|---|
| $\underline{x}^o$ = (0.9998 0.99969 0.99974 0.99933 0.99928 0.9991 0.998 0.99572 0.99137 0.98285) |

The last two testing functions are extremely challengeable. Table 3.4.6 summarizes the two testing functions testing results. The lambda algorithm searching scheme utilizes (5, 4,120) and (5, 4, 400) respectively.

Table 3.4.6 Algorithm efficiency indices for 30-dimensional Rosenbrock function and 100-dimensional sin20 function

| Search indices | Rosenbrock | $\sin^{20}$ |
|---|---|---|
| Domain | $[-2.408, 2.408]^{30}$ | $[-10,10]^{100}$ |
| Time (sec.) | 90.95 | 291.06 |
| Loop | 100 | 100 |
| Probability control | 0.60 | 0.3 |

**8. Rosenbrock Function:** Number of Variables: *n* =30, the minimum is 0 when $x_i = 1$, $i = 1,\cdots,30$. The string configuration for searching lambda scheme is (5, 4,120).

The searched minimum = 2.5183.

| |
|---|
| Optimal $\underline{x}^o$ =(0.9994 0.9973 0.9994 1.0024 1.0014 1.0003 1.0035 1.0007 1.0015 1.0007 1.0004 1.0015 0.9971 0.9966 1.0001 0.9995 0.9973 0.9997 0.9952 0.9991 0.9927 0.9856 0.9707 0.9473 0.9014 0.9520 0.9744 0.9714 0.9491 0.9008) |

**9.** $\sin^{20}$ **function:** Number of Variables: *n* =100. The theoretical maximum is *n* when $x_i = j\pi + \pi/2, \quad j = 0,1,2,\cdots$.

$$f(x_1,\cdots,x_n)=\sum_{i=1}^{n}\sin^{20}(x_i) \tag{35}$$

The searched maximum value = 91.0671, the optimal

$\underline{x}^o$ =(1.4857 -7.8411 -4.6816 -4.7119 -1.5208 -4.6401 1.5676 7.8522 1.6444 7.8734 -
7.8784 4.7178 -1.5417 -7.8683 1.5590 7.7531 7.8484 -1.5301 -4.7494 -4.8971 1.5430
-4.7074 -1.3356 -1.5826 4.7689 2.9632 -1.5465 1.6441 -1.5685 1.4859 1.5135 1.5899
1.5212 0.1591 -0.9106 1.5329 -4.7451 4.6183 4.7783 7.8759 1.5594 1.5520 -4.8942 -
1.5857 -1.5533 4.6714 1.5529 1.6801 1.5276 1.5989 -7.8975 -7.8473 -4.7436 -1.5183
4.7857 7.8521 4.7119 -1.5925 -1.6175 -1.6763 4.7724 -4.7648 -4.6491 1.5808 -7.8505
-7.8527 4.6484 3.7575 4.7969 -1.5098 1.4933 7.8250 1.7861 -1.4387 -1.5806 7.8622 -
1.4867 -4.6982 -4.3371 1.5879 1.7464 -1.4736 -7.8447 1.5872 1.5873 -1.5733 -7.9322
4.7021 1.5871 1.5891 -4.7056 1.6080 4.8077 -1.5593 1.5689 -4.7446 -1.5889 -1.4475
-4.7321 1.7987)

In summary, the algorithm testing demonstrates satisfactory result in accuracy and efficiency.

### 3.4.7 Likelihood-lambda procedure

Likelihood function and procedure plays important role in safety and reliability modelling, see Ben-Gal (2007), Lawless (1982), Ushakov (1994). In this section, we will investigate the scheme to utilize the lambda algorithm for searching the numerical solution to a likelihood function.

#### 3.4.7.1 Log-likelihood function

Let $L(\underline{\theta}|\mathrm{K}) = f(x_1, \vartheta_1, x_2, \vartheta_2, \cdots, x_N, \vartheta_N; \underline{\theta})$ with $f(\cdot;\underline{\theta})$ representing the joint distribution of data $\mathrm{K}$ . This is then called the likelihood function with respect to parameter set $\underline{\theta}$ , $\underline{\theta} \in \Theta$ .

**Definition 22:** Let $\mathrm{K} = \{(x_i, \vartheta_i), i = 1, 2, \cdots, N\}$ be a failure-censoring data record, i.e.

$$\vartheta_i = \begin{cases} 0 & x_i \text{ is a natural failure} \\ 1 & x_i \text{ is a censored event} \end{cases} \tag{36}$$

then

$$L(\underline{\theta}|\mathrm{K}) = \prod_{i=1}^{N} f^{1-\vartheta_i}(x_i;\underline{\theta})\, R^{\vartheta_i}(x_i;\underline{\theta}) \tag{37}$$

where $f$ is the failure density function and $R$ is the reliability function.

**Definition 23:** The function then:

$$l(\underline{\theta}|\mathrm{K}) = \ln\left(L(\underline{\theta}|\mathrm{K})\right) \tag{38}$$

is called the log-likelihood function.

**Lemma 24:** $\underline{\theta}_0$ is an optimal point for $l(\underline{\theta}\,|\,\mathrm{K})$ if and only if it is an optimal point for $L(\underline{\theta}\,|\,\mathrm{K})$. Note that $\ln(\cdot)$, whose base is $e>1$, is monotone increasing. Therefore the patterns in $L(\underline{\theta}\,|\,\mathrm{K})$ will be well maintained by $l(\underline{\theta}\,|\,\mathrm{K})$ and the converse is also true: then

$$\begin{aligned} &l(\underline{\theta}_0\,|\,\mathrm{K}) = \max\left\{l(\underline{\theta}\,|\,\mathrm{K})\right\} \\ &\Leftrightarrow \\ &L(\underline{\theta}_0\,|\,\mathrm{K}) = \max\left\{L(\underline{\theta}\,|\,\mathrm{K})\right\} \end{aligned} \tag{39}$$

Turning our attention now to wave-like lifetime distribution of Type I, (see [30], [31]), it has a form:

$$F(x) = 1 - \exp\left(-\int_0^x \left(\gamma + \frac{\sin^2 \alpha s}{s^2}\right) ds\right) \tag{40}$$

with two-parameter hazard function:

$$\begin{aligned} &h(x) = \gamma + \frac{\sin^2 \alpha x}{x^2} \\ &x \in [0, +\infty),\ \alpha > 0,\ \gamma \geq 0 \end{aligned} \tag{41}$$

**Theorem 25:** (Guo et al., 2010) For the Type I wave-like distribution, the log-likelihood function is:

$$\begin{aligned} l(\alpha,\gamma\,|\,\mathrm{K}) = &\sum_{i=1}^{N} (1-\vartheta_i) \ln\left(\gamma + \frac{\sin^2 \alpha x_i}{x_i^2}\right) \\ &- \sum_{i=1}^{N} \int_0^{x_i} \left(\gamma + \frac{\sin^2 \alpha s}{s^2}\right) ds \end{aligned} \tag{42}$$

The first-order partial derivatives are

$$\begin{aligned} \frac{\partial l(\alpha,\gamma\,|\,\mathrm{K})}{\partial \alpha} &= \sum_{i=1}^{N} \frac{\sin(2\alpha x_i)(1-\vartheta_i)\, x_i}{\gamma x_i^2 + \sin^2 \alpha x_i} \\ &\quad - \sum_{i=1}^{N} \int_0^{x_i} \frac{\sin(2\alpha s)}{s} ds \\ \frac{\partial l(\alpha,\gamma\,|\,\mathrm{K})}{\partial \gamma} &= \sum_{i=1}^{N} \frac{x_i^2 (1-\vartheta_i)}{\gamma x_i^2 + \sin^2 \alpha x_i} - \sum_{i=1}^{N} x_i \end{aligned} \tag{43}$$

and the second-order order partial derivatives are

$$\begin{aligned}\frac{\partial^2 l(\alpha,\gamma\,|\,\mathrm{K})}{\partial^2\alpha} &= \sum_{i=1}^{N}\frac{2\cos(2\alpha x_i)\left(\gamma x_i^2+\sin^2\alpha x_i\right)-\sin^2(2\alpha x_i)}{\left(\gamma x_i^2+\sin^2\alpha x_i\right)^2}(1-\vartheta_i)x_i^2\\ &\quad -\frac{1}{\alpha}\sum_{i=1}^{N}\sin(2\alpha x_i)\\ \frac{\partial^2 l(\alpha,\gamma\,|\,\mathrm{K})}{\partial\alpha\partial\gamma} &= -\sum_{i=1}^{N}\frac{x_i^3(1-\vartheta_i)\sin(2\alpha x_i)}{\left(\gamma x_i^2+\sin^2\alpha x_i\right)^2}\\ \frac{\partial^2 l(\alpha,\gamma\,|\,\mathrm{K})}{\partial\gamma^2} &= -\sum_{i=1}^{N}\frac{x_i^4(1-\vartheta_i)}{\left(\gamma x_i^2+\sin^2\alpha x_i\right)^2}\end{aligned} \tag{44}$$

**Theorem 26:** For the Type II wave-like lifetime distribution with 2 parameters, and a hazard function of the form $h(x)=\gamma+\sin(\alpha x)/x$, the log-likelihood function in the presence of both failures and censored data is

$$\begin{aligned}l(\alpha,\gamma\,|\,\mathrm{K}) &= \sum_{i=1}^{N}(1-\vartheta_i)\ln\left(\gamma+\frac{\sin(\alpha x_i)}{x_i}\right)\\ &\quad -\gamma\sum_{i=1}^{N}x_i-\sum_{i=1}^{N}\int_0^{x_i}\frac{\sin(\alpha s)}{s}ds\end{aligned} \tag{45}$$

The first-order partial derivatives are

$$\begin{aligned}\frac{\partial l(\alpha,\gamma\,|\,\mathrm{K})}{\partial\alpha} &= \sum_{i=1}^{N}(1-\vartheta_i)\frac{x_i\cos(\alpha x_i)}{\gamma x_i+\sin(\alpha x_i)}\\ &\quad +\frac{1}{\alpha}\sum_{i=1}^{N}\sin(\alpha x_i)\\ \frac{\partial l(\alpha,\gamma\,|\,\mathrm{K})}{\partial\gamma} &= \sum_{i=1}^{N}(1-\vartheta_i)\frac{x_i}{\gamma x_i+\sin(\alpha x_i)}-\sum_{i=1}^{N}x_i\end{aligned} \tag{46}$$

and the second-order partial derivatives are

$$
\begin{aligned}
\frac{\partial^2 l(\alpha,\gamma \mid \mathrm{K})}{\partial \alpha^2} &= -\sum_{i=1}^{N}(1-\vartheta_i)\,x_i^2\,\frac{\gamma x_i \sin(\alpha x_i)+1}{\left(\gamma x_i+\sin(\alpha x_i)\right)^2} \\
&\quad -\frac{1}{\alpha^2}\sum_{i=1}^{N}\sin(\alpha x_i)+\frac{1}{\alpha}\sum_{i=1}^{N}x_i\cos(\alpha x_i) \\
\frac{\partial^2 l(\alpha,\gamma \mid \mathrm{K})}{\partial \alpha \partial \gamma} &= -\sum_{i=1}^{N}\frac{(1-\vartheta_i)\,x_i^2\cos(\alpha x_i)}{\left(\gamma x_i+\sin(\alpha x_i)\right)^2} \\
\frac{\partial^2 l(\alpha,\gamma \mid \mathrm{K})}{\partial \gamma^2} &= -\sum_{i=1}^{N}\frac{(1-\vartheta_i)\,x_i^2}{\left(\gamma x_i+\sin(\alpha x_i)\right)^2}
\end{aligned}
\tag{47}
$$

**Remark 27:** Theorem 25 and 26 facilitate classical maximum likelihood estimation with derivatives up to the second order for the two types of wave-like lifetime distributions. Reliability engineers can use these two theorems for modeling and analysis in traditional Newton-Raphson procedure or use semi-derivative or non-derivative Likelihood-GA procedure if they do not mind the computation time consumptions. To reach a better efficiency, we intend to switch our attention into replacing the GA part by lambda algorithm.

### 3.4.7.2 Log-likelihood function

The ML-lambda procedure for searching solutions to the joint non-linear equation system:

$$
\begin{cases}
\partial l(\alpha,\gamma \mid \mathrm{K})/\partial \alpha = 0 \\
\partial l(\alpha,\gamma \mid \mathrm{K})/\partial \gamma = 0
\end{cases}
\tag{48}
$$

because the integral term appears in the wave-like log-likelihood function. The searching results for the two models are listed in Table 3.4.7.

Table 3.4.7 The MLE of parameters for wave-likelihood lifetime distributions

| Type | I | II |
|---|---|---|
| $\hat{\alpha}$ | 6.5202 (0.00169) | 0.0412 (0.01310) |
| | | 0.0961 (0.0006961) |
| $\hat{\gamma}$ | 0.0001 (0.00001) | 0.0001 (0.00001) |
| | | 0.0206 (0.0000085) |

| | | |
|---|---|---|
| $l(\hat{\alpha},\hat{\gamma}\mid K)$ | -3293.1074 | -1719.2372 |
| | | -36496.9421 |
| Accuracy | 5.5807e-008 | 2.5757e-008 |
| | | 2.1534e-006 |
| Computation time | 17.9384 sec. | 17.0387 sec. |
| | | 74.9782 sec. |

In the Table 3.4.7, for the parameter estimate columns, the top figures are the estimators whereas the figures in brackets are estimated standard deviations.

It is observed that in the case of Type I, the first pair gives the local maximum ($l(\hat{\alpha},\hat{\gamma}\mid K)$= -3293.1074), the second is a global ($l(\hat{\alpha},\hat{\gamma}\mid K)$= -1719.2372), whereas one suspects that the Type II model is a better description of the failure/repair process in operation here ($l(\hat{\alpha},\hat{\gamma}\mid K)$= -36496.9421). We found two optimal solutions for Type II model (blue is the first, black is the second). The following three figures plot the estimated hazard functions and e-score plots.

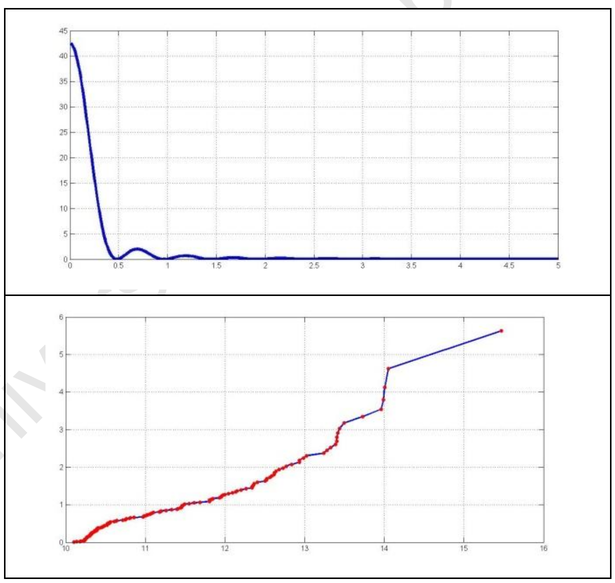
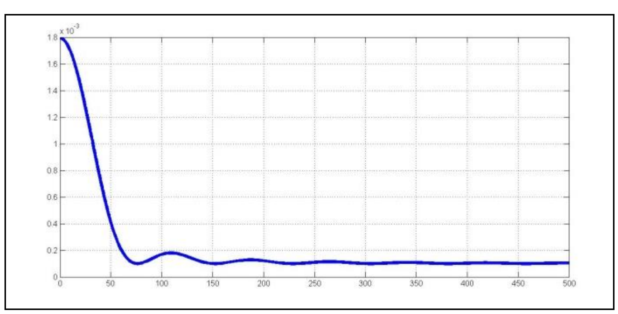


Figure 3.4.6 The estimated hazard function of Type I wave-like lifetime distribution ($\hat{\alpha}$=6.5202, $\hat{\gamma}$=0.0001) and approximated e-score plot



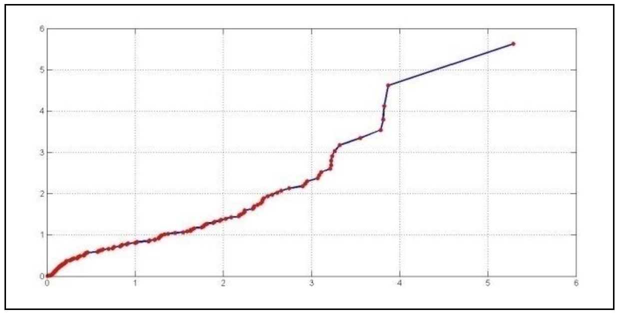


Figure 3.4.7 The estimated hazard function of Type I wave-like lifetime distribution ($\hat{\alpha}$ =0.0412, $\hat{\gamma}$ = 0.0001) and approximated e-score plot

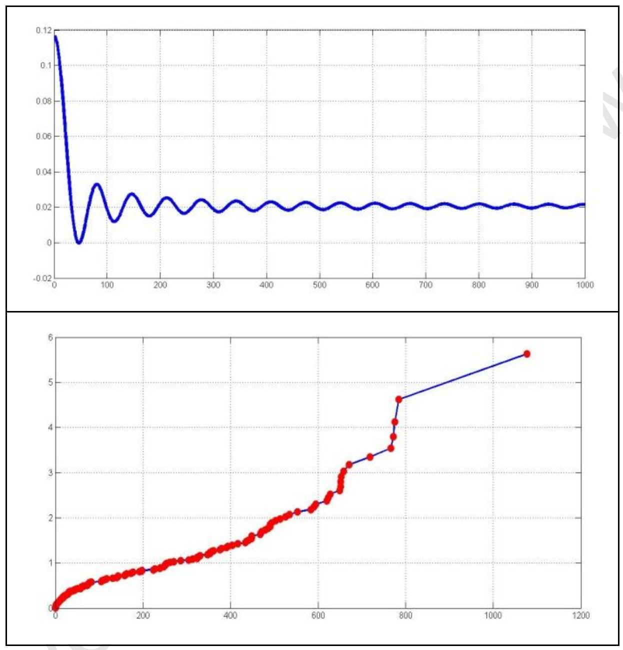


Figure 3.4.8 The estimated hazard function of Type II wave-like lifetime distribution ( $\hat{\alpha}$ =0.0961, $\hat{\gamma}$ =0.0206) and approximated e-score plot

**Remark 28:** The e-score plot (Lawless, 1982) is based on a fact that

$$\hat{e}_i = \int_0^{x_i} h\left(s;\hat{\alpha},\hat{\gamma}\right) ds \overset{d}{\sim} \exp\left(-x_i\right) \tag{49}$$

and

$$\mathrm{E}\left[\hat{e}_{(i)}\right] = \sum_{l=1}^{i} \frac{1}{n-l+1} \tag{50}$$

where $\hat{e}_{(i)}$ is the $i^{th}$ order statistic in calculated e-scores $\{\hat{e}_1, \hat{e}_2, \cdots, \hat{e}_N\}$ . E-score plot plots ( $\hat{e}_{(i)}$ , $\mathrm{E}\left[\hat{e}_{(i)}\right]$), $i = 1, 2, \cdots, N$ . If the plot demonstrates a straight-line then the good-fitness of the maximum likelihood is good enough. From the three e-score plots, we see similar patterns, but Type I model global result in Figure 3.4.7 ( $\hat{\alpha}$ =0.0412, $\hat{\gamma}$ = 0.0001) convinces us more.

### 3.4.8 Conclusion

In this paper, we introduce the new lambda algorithm first, and then investigate the underlying operating mechanism of the lambda algorithm. Furthermore, we explore the merging the lambda algorithm with maximum likelihood procedure. We have a detailed illustrative application. In the future, we will strive to explore more safety and reliability applications.

# 3.5 Decision Theory under General Uncertainty

Yanhong Cui, Renkuan Guo, Tim Dunne

Department of Statistical Sciences, University of Cape Town

Private Bag, Rhodes' Gift, Rondebosch 7701, Cape Town, South Africa

hadisna@hotmail.com renkuan.guo@uct.ac.za tim.dunne@uct.ac.za

Danni Guo

South African National Biodiversity Institute,

Private Bag X7, Claremont 7735, Cape Town, South Africa

d.guo@sanbi.org.za



This paper has been published in Summer Safety and Reliability Seminars, SSARS 2010, Journal of Polish Safety and Reliability Association, Vol. 1, pp 51 – 66. 2010. ISBN: 978-83-925436-2-6. Editors in Chief: Krzysztof Kolowrocki, Joanna Soszynska and Enrico Zio.

### 3.5.1 Abstract

The exposure of Toyota management's cover-up of its faulty car component problems raises a fundamental question: did Toyota management make an appropriate decision taking all uncertainties into account? Statistical decision theory is a framework with a probabilistic foundation, which admits random uncertainty about the real world and human thinking. In general, the uncertainty of the real world is diversified and therefore the effort of trying to deal with different forms of uncertainty with one special form of uncertainty, namely random uncertainty, may be oversimplified. In this paper, we introduce an axiomatic uncertain measure theoretical framework and explore the essential mechanism in formulating a general uncertainty decision theory. We expect that a new understanding of uncertainty and development of a corresponding new uncertainty decision-making approach may assist intelligence communities to survive and deal with the extremely tough and diverse aspects of an uncertain reality.

**Keywords:** statistical decision, state space, action space, uncertain measure, loss function, risk, uncertainty decision

### 3.5.2 Introduction

Recently a "Made in Japan" crisis began spreading widely, triggered by a Toyota Prius brake fault, a fire-sparking Honda Jazz electric-window, and a Sony Camera problem, and a cover-up of faulty seats for Boeing 747 jets. These events have shocked worldwide business and industry.

Journalism today has largely castigated the Toyota decision makers for censoring and even hiding fault factors from the public, particularly, attempting to prevent release to the press. In some sense, the journalists are correct, a posterior, but in the decision point of view, they are not necessarily correct a priori, because not all the "factors" (states of nature) of car making necessarily enter into the decision mechanisms – the model distributions, as when some states are allocated a tiny possibility or ignored.

Furthermore, active members of safety and reliability communities may do well to calmly re-examine whether or not the theoretical roots and foundation of statistical decision theory are sufficient for the purpose to which it is applied. Randomness is merely one of the forms of uncertainty. In consequence, existing statistical theory may not always provide a completely suitable analysis for data embodying more general uncertainty.

The real world is not as simple as we imagine. Uncertainty is intrinsic and diversified in form. For example, vagueness is a different form of uncertainty from randomness, and enters more and more into today's industrial environments, as Carvalho and Machado (2006) have commented, "In a global market, companies must deal with a high rate of changes in business environment. … The parameters, variables and restrictions of the production system are inherently vagueness." Therefore one may argue that a company decision should no longer only be a somewhat routine exercise of applying traditional statistical decision techniques arising from its sound but constrained probabilistic underpinnings. Without a thorough and explicit exploration of uncertainty and its characteristics, attempts to abstract real world uncertainty into appropriate concepts will inevitably permit that decision exercises to fall short of the reality of business. Our "reality", its diversity, and the formality of a general uncertainty are the fundamental rationale for us to pursue the exploration of a new general uncertainty decision approach.

Decision making is based on a consensus "truth", whether or not that truth exists in the eyes of other communities. Furthermore, some factors, such as the faulty seat phenomena and cover-up are not repeatable incidents, and cannot admit complete probability assignment, i.e., assigning probability to such an isolated incident is illogical, but, top managements must include such "accident" factors in decision-making. Thus, it is logical to say decision is a subjective activity.

The discipline of Statistics builds upon probability theory and deals with collecting, analysing, and drawing conclusions from data information essentially featured by random uncertainty and imposed upon modelled pattern (in terms of a probability distribution). It is essential to emphasize that the mechanism underlying statistical decision is the use of probability distributions. An uncertainty decision problem is essentially the appropriate specification of uncertain distributions.

The fundamental problem here is what uncertainty distributions one may invoke to characterize the relevant states and events. Recently, Professor Liu from Tsinghua University (Beijing, China) proposed an axiomatic uncertainty measure theory (2010), which is sub-$\sigma$-additive and less restrictive than the ($\sigma$-additive) probability measure. A $\sigma$-algebra, denoted by $\mathfrak{A}(\Xi)$ is a collection of subsets (events) in a set $\Xi$ satisfying three properties: (i) $\Xi \in \mathfrak{A}(\Xi)$; (ii) $\forall A \in \mathfrak{A}(\Xi)$, $A^c \in \mathfrak{A}(\Xi)$, where $A^c$ is the complement of $A$ in $\Xi$; (iii) $\bigcup_{n=1}^{+\infty} A_n \in \mathfrak{A}(\Xi)$ Probability measure and uncertain measure may be defined on a $\sigma$-algebra, and each characterises a probability distribution or a uncertainty distribution respectively. By virtue of its less restrictive nature, Liu’s (2010) uncertain measure theory can support uncertainty distribution building, analysis and modelling more general uncertainty observations and their use in the making of a decision.

The remainder of the paper is structured as follows: Section 3.5.3 introduces Liu’s (2007, 2010) new axiomatic uncertain measure theory, which permits the new uncertainty distribution underlying the decision mechanism; Section 3.5.4 reviews the (probabilistic) statistical decision theory in order to reveal the underlying mechanism behind statistical decision making – relevant (probability) distributions. This review suggests that the new general uncertainty decision approach should preserve the basic framework of the (probabilistic) statistical decision theory but in some conditions replace the underlying probability distributions with appropriate uncertain distributions. Section 3.5.5 discusses the basic elements and some intrinsic features of uncertainty decision theory in comparisons with statistical decision theory. Sections 3.5.6 and 3.5.7 illustrate the general uncertainty decision making approach in discrete and continuous uncertainty environments respectively; Section 3.5.8 concludes the paper.



### 3.5.3 Uncertain measure foundation

Uncertain measure (Liu, 2010) is an axiomatically defined set function mapping from a $\sigma$-algebra of a given space (set) to the unit interval [0, 1], which provides a measuring grade system for an uncertain phenomenon and permits the formal definition of an uncertain variable.

Let $\Xi$ be a nonempty set (space), and $\mathfrak{A}(\Xi)$ the $\sigma$-algebra on $\Xi$. Each subset $A \subset \Xi$, $A \in \mathfrak{A}(\Xi)$ is called an uncertain event. A number denoted $ƛ\{A\}$, $0 \le ƛ\{A\} \le 1$, is assigned to event $A$, which indicates the uncertain measuring grade with which event $A$ occurs. Occurrence of an event A is defined as occurrence of any constituent outcome x within $A$. The set function $ƛ\{A\}$ satisfies the following axioms given by Liu (2010):

**Axiom 1:** (Normality) $ƛ\{\Xi\} = 1$.

**Axiom 2:** (Monotonicity) $ƛ\{\cdot\}$ is non-decreasing: If $A \subset B$, then $ƛ\{A\} \le ƛ\{B\}$.

**Axiom 3:** (Self-Duality) $ƛ\{\cdot\}$ is self-dual, If $A \in \mathfrak{A}(\Xi)$, then $ƛ\{A\} + ƛ\{A^c\} = 1$, where $A^c$ is the complement of $A$ in $\Xi$.

**Axiom 4:** ($\sigma$- Subadditivity) $ƛ\left\{\bigcup_{i=1}^{\infty} A_i\right\} \le \sum_{i=1}^{\infty} ƛ\{A_i\}$ for any countable event sequence $\{A_i\}$.

**Definition 1:** (Liu, 2010) Any set function $\lambda : \mathfrak{A}\ \Xi \rightarrow 0,1$ satisfying Axioms L.1-L.4 is called an uncertain measure. The triple $\Xi, \mathfrak{A}\ \Xi\ , \lambda$ is called an uncertain space.

**Definition 2:** (Liu, 2010) An uncertain variable $\xi$ is a measurable mapping, i.e., $\xi : (\Xi, \mathfrak{A}(\Xi)) \rightarrow (\mathbb{R}, \mathfrak{B}(\mathbb{R}))$, where $\mathfrak{B}(\mathbb{R})$ is the Borel $\sigma$-algebra on $\mathbb{R} = (-\infty, +\infty)$.

**Remark 3:** The Borel $\sigma$-algebra is the smallest set class of Borel sets on $\mathbb{R} = (-\infty, +\infty)$. The Borel sets include $\mathbb{R} = (-\infty, +\infty)$, empty set $\varnothing$, all the closed intervals, $[a,b]$, all the semi-closed intervals, $[a,b)$, $(a,b]$, $[a,+\infty)$, and $(-\infty,b]$, and all the open intervals, $(a,b)$, $(-\infty,b)$, and $(a,+\infty)$, where $\forall a,b \in \mathbb{R}$, $-\infty < a \leq b < +\infty$.

**Remark 4:** The fundamental difference between a random variable and an uncertain variable is the measure space on which they are defined. In the associated triples, the first two elements are similar in form: the set and a $\sigma$-algebra on the set. However, the third elements in the triples: the measures defined on the $\sigma$--algebras, are not similar. The former (i.e. the probability measure Pr) obeys $\sigma$-additivity and the later (i.e. the uncertain measure $\lambda$ obeys only $\sigma$-subadditivity. The choice of a measure inevitably has impacts on the behaviour of any measurable function on the triple.

**Definition 5:** (Liu, 2010) The uncertain distribution $\Psi : \mathbb{R} \rightarrow [0,1]$ of an uncertain variable $\xi : \Xi \rightarrow \mathbb{R}$ defined on the uncertain space $\Xi, \mathfrak{A}\ \Xi\ , \lambda$ is

$$\Psi(x) = \lambda\{\tau \in \Xi | \xi(\tau) \leq x\} \tag{1}$$

**Remark 6:** A random variable $X$ is a measurable mapping. To understand the measurability of a random variable, particularly, the role played by the $\sigma$-algebra $\mathfrak{F}(\Omega)$, we note how measurability is structured for a random variable. Let $(\Omega, \mathfrak{F}, P)$ be a probability space and $(\mathbb{R}, \mathfrak{B}(\mathbb{R}))$ be a measurable space on real-line, then a real-valued function $X$ is random variable if and only if the pre-image $\{\omega \in \Omega : X(\omega) \leq r\} \in \mathfrak{F}$, for all $r \in \mathbb{R}$,. For each value $r \in \mathbb{R}$ taken by a real-valued random variable $X$, the event $B = (-\infty, r]$ is an element of the Borel $\sigma$-algebra over a real-line $\mathbb{R}$, the pre-image of event $B$ under random variable $X$ is an event

$$\omega \in \Omega : X\ \omega\ \in B\ =\ \omega \in \Omega : X\ \omega\ \leq r \tag{2}$$

where $\{\omega \in \Omega : X(\omega) \leq r\}$ is an element of $\sigma$- algebra $\mathfrak{F}$ over $\Omega$, and the probability measure $P$ is defined on this set class, i.e., $\sigma$- algebra $\mathfrak{F}$, i.e., $P : \mathfrak{F} \rightarrow [0,1]$. Therefore every element (event) of $\mathfrak{F}$ is assigned with a probability grade, i.e., event $\{\omega \in \Omega : X(\omega) \leq r\}$ is assigned a probability grade, which is $P\{\omega \in \Omega : X(\omega) \leq r\}$.

Thus, $\sigma$-algebra $\mathfrak{F}$ facilitates the formal definition of a random variable in terms of the membership of the pre-image $\{\omega \in \Omega : X(\omega) \leq r\}$ within the $\sigma$- algebra $\mathfrak{F}$, on which the probability measuring grade defined. Every event in $\sigma$-algebra $\mathfrak{F}$ is assigned a probability. Each random variable on the probability space $(\Omega, \mathfrak{F}, P)$ induces a probability space $(\mathbb{R}, \mathfrak{B}(\mathbb{R}), \mu)$ by means of the following well-known correspondence.

$$\forall B \in \mathfrak{B}(\mathbb{R}) : \mu(B) = P(X^{-1}(B)) = P(X \in B) \tag{3}$$

Let us write $\mu = P \circ X^{-1}$ and specifically, the probability distribution is defined by the induced measure $\mu$ ,

$$F(r) = \mu((-\infty, r]) = P(X \leq r) \tag{4}$$

In all, the random variable $X$ defined on a given probability space $(\Omega, \mathfrak{F}(\Omega), P)$ is a measurable mapping to $(\mathbb{R}, \mathfrak{B}(\mathbb{R}))$ and thus induces the distribution function, $F : \mathbb{R} \rightarrow [0,1]$, which is used to characterize the random variable.

Similarly, in the axiomatic development of uncertain measure, the $\sigma$-algebra $\mathfrak{A}(\Xi)$ plays critical roles as the set class in defining both the measurability of an uncertain variable $\xi$ and the set function $\lambdabar$ as the uncertain measure. The roles are equivalent to the roles played by a $\sigma$-algebra in probability measure theory in defining both the measurability of a random variable $X$ and the set function $P$ as the probability measure. As long as an uncertain measure $\lambdabar$ is specified, the uncertain distribution $\Psi$ is fully defined. The next theorem states the necessary and sufficiency conditions for a function to be an uncertain distribution.

**Theorem 7:** Let $\Psi : \mathbb{R} \rightarrow [0,1]$ be a non-decreasing function with

$$\Psi(-\infty) = 0, \Psi(+\infty) = 1. \tag{5}$$

Then set function $\nu : \mathfrak{B}(\mathbb{R}) \rightarrow [0,1]$, for any Borel set $B$ :

$$\nu\{B\}=\begin{cases}\nu^*\{B\} & \text{if } \nu^*\{B\}<0.5\\ 1-\nu^*\{B^c\} & \text{if } \nu^*\{B^c\}<0.5\\ 0.5 & \text{otherwise}\end{cases} \tag{6}$$

where

$$\nu^*\{B\}=\nu_1\{B\}\wedge\nu_2\{B\}\wedge\nu_3\{B\},\forall B\in\mathfrak{B}(\mathbb{R}) \tag{7}$$

with $\nu_i:\mathfrak{B}(\mathbb{R})\to[0,1]$, $i=1,2,3$, given by

$$\nu_1\{B\}=\begin{cases}1-\lim\limits_{x\uparrow\inf\{B\}}\Psi(x) & \text{if } \inf\{B\}\in B\\ 1-\Psi(\inf\{B\}) & \text{otherwise}\end{cases}, \tag{8}$$

and

$$\nu_2\{B\}=\Psi(\sup\{B\}), \tag{9}$$

and

$$\nu_3\{B\}=\inf_{(a,b]\subset B^c}\{\Psi(a)+1-\Psi(b)\} \tag{10}$$

is an uncertain measure on the Borel $\sigma$-algebra, $\mathfrak{B}(\mathbb{R})$

**Remark 8:** Technically, once the induced uncertain measure $\nu$ is defined by Eq. (6), the uncertain space $(\mathbb{R},\mathfrak{B}(\mathbb{R}),\nu)$ is established. Then the mapping from $(\mathbb{R},\mathfrak{B}(\mathbb{R}),\nu)$ to $(\Xi,\mathfrak{A}(\Xi),\lambda)$ can be built, i.e., the uncertain measure $\lambda$ is fully specified.

For comparison purposes, we note the definition of the probability distribution.

**Definition 9:** Let $(\Omega,\mathfrak{F}(\Omega),P)$ be a given probability space, the probability distribution of a random variable on $(\Omega,\mathfrak{F}(\Omega),P)$ is

$$F(x)=P\{\omega\in\Omega|\,X(\omega)\le x\} \tag{11}$$

**Theorem 10:** Let $F:\mathbb{R}\to[0,1]$. Then, $F$ is a probability distribution if and only if $F$ satisfies each of the following three conditions:

( i) $\lim\limits_{x\to-\infty}F(x)=0, \lim\limits_{x\to+\infty}F(x)=1$;

(ii) $F(x)$ is non-decreasing in x;

(iii) $F$ is right-continuous, i.e., $\forall x_0\in\mathbb{R}$, $\lim\limits_{x\downarrow x_0}F(x)=F(x_0)$

**Remark 11:** The difference between a probability distribution and an uncertainty distribution relates to whether the distribution possesses right-continuity. The relaxation of the condition in the uncertain distribution function arises from the sub-$\sigma$-additivity property of the underlying uncertain measure $\lambda$. On the basis of this distributional difference, the new uncertainty decision theory developed in this paper differs from statistical decision theory without relying on any arguments about any interpretation differences of the two distributions.

**Definition 12:** (Liu, 2010) An n-dimensional uncertain vector from an uncertain measure space $\left(\Xi, \mathfrak{A}(\Xi), \lambda\right)$ to the set of n-dimensional real-valued vectors, i.e., for Borel set $B$ within $\mathbb{R}^n$, the set

$$\left\{\underline{\xi} \in B\right\} = \left\{\tau \in \Xi \middle| \underline{\xi}(\tau) \in B\right\} \tag{12}$$

is an event.

**Theorem 13:** (Liu, 2010) Let $\underline{\xi} = \left(\xi_1, \xi_2, \cdots, \xi_n\right)^T$ be an uncertain vector, and $f : \mathbb{R}^n \to \mathbb{R}$ a measurable function. Then $f\left(\underline{\xi}\right)$ is an uncertain variable such that

$$\lambda\left\{f\left(\underline{\xi}\right) \in B\right\} = \lambda\left\{\underline{\xi} \in f^{-1}\left(B\right)\right\} \tag{13}$$

for any Borel set $B$ within $\mathbb{R}^n$.

Now, we are ready to investigate statistical decision theory and extend its principles to uncertainty decision theory because we have adequate self-contained materials to understand the further explorations.



### 3.5.4 Statistical Decision Theory

Statistical decision theory is established on the axiomatic foundation of probability measure, see Kolmogorov (1950) and Primas (1999). The developments can be sourced in DeGroot (1970) and Ferguson (1967). A measure theoretical decision theory is stated below using well-know results.

#### 3.5.4.1 Three elements of statistical decision

The three elements of statistical decision theory are: the states of the nature, the action space, and the loss in elementary statistical course. At the measure theoretical level, in order to simplify the mechanism underlying statistical decision, we use “sample space” instead of “states of nature”. Note here the term “statistical decision” implies that a decision is made by observation-based statistical analysis. Without data and the distribution underlying the data, there is nothing. While both “sample space and distributional family” and “states of nature” may share the same meanings, the former is more comprehensive and characteristic-exposing and the latter is more intuitive. In discussing statistical decision theory we use “sample space and distributional family” to emphasize the observational and data oriented statistical nature of the set for specifying a decision problem.

(1) **Sample space and distributional family**. In measure theoretical language, every concrete value of $X$, denoted as $x$, is called a sample value. The set of all possible sample values contribute to a sample space, denoted as $\mathfrak{X} = \{x, x \in X\}$. It is necessary to emphasize that the specification of $\mathfrak{X}$ only requires $\mathfrak{X}$ contains all the possible values of $X$, but it is not required that $X$ must admit all values in $\mathfrak{X}$. For example, though $X$ may take non-negative real values, i.e., $X = \mathbb{R}^+ = [0, +\infty)$ we may nonetheless introduce the sample space as $\mathfrak{X} = \mathbb{R} = (-\infty, +\infty)$. This property may provide great conveniences in mathematical treatments later. Furthermore, a $\sigma$-algebra (field) is specified on $\mathfrak{X}$, denoted as $\mathfrak{B}(\mathfrak{X})$. In statistical inferences, in specifying sample space, it is necessary to specify both $\mathfrak{X}$ and $\mathfrak{B}(\mathfrak{X})$, i.e., the measurable space $(\mathfrak{X}, \mathfrak{B}(\mathfrak{X}))$. Thus, in measure theoretical treatments, it is often the practice to regard the measurable space $(\mathfrak{X}, \mathfrak{B}(\mathfrak{X}))$ as a sample space, instead of simply $\mathfrak{X}$ as in elemental statistics. The most commonly used sample space $\mathfrak{X}$ is the n-dimensional Euclidean space $\mathbb{R}^n$, while $\mathfrak{B}(\mathfrak{X})$ is the $\sigma$-algebra of the Borel sets in $\mathbb{R}^n$, or $\mathfrak{X}$ is a subset of the n-dimensional Euclidean space $\mathbb{R}^n$, while $\mathfrak{B}(\mathfrak{X})$ is the $\sigma$-algebra of the Borel sets of $\mathfrak{X}$. Often there is no rigorous distinction between these two forms. Without special claims, when using $(\mathfrak{X}, \mathfrak{B}(\mathfrak{X})) = (\mathbb{R}^n, \mathfrak{B}(\mathbb{R}^n))$, the later cases are covered. $(\mathbb{R}^n, \mathfrak{B}(\mathbb{R}^n))$ is called the Euclidean sample space.

On the $\sigma$-algebra $\mathfrak{B}(\mathfrak{X})$, a family of probability measures $\{P_\theta, \theta \in \Theta\}$ is defined, where $\Theta$ is called a parameter space, and in many situations, $\Theta \subset \mathbb{R}^m$. The distribution of $X$ is one of the members in the distributional family $\{P_\theta, \theta \in \Theta\}$, i.e., there exists a $\theta_0 \in \Theta$, such that the distribution of $X$ is $P_{\theta_0}$ but the value of $\theta_0$ is unknown. Determining a value of $\theta_0$, i.e., the specific distribution for $X$ from the distributional family $\{P_\theta, \theta \in \Theta\}$ is precisely the object of the statistical inference.

The sample space and distributional family together determine the probability mechanism of the observations (i.e., sample values) from the population $X$. This pair is often written in the form of $\{(\mathfrak{X}, \mathfrak{B}(\mathfrak{X}), P_\theta), \theta \in \Theta\}$ or alternatively, we say that the sample space of $X$ is $(\mathfrak{X}, \mathfrak{B}(\mathfrak{X}))$ with distributional family $\{P_\theta, \theta \in \Theta\}$.

Recall that $(\Omega, \mathfrak{A}, P)$ is called a probability space, therefore, it is important to bring the family of the probability spaces $\{(\mathfrak{X}, \mathfrak{B}(\mathfrak{X}), P_\theta), \theta \in \Theta\}$ into the decision process since the selection of $\theta$ is the basic implicit task.

(2) **Decision space**. Statistical decision making (or inference), whether it takes the form of point estimation, or interval estimation, or hypothesis testing, is actually decision making based on the sample information (statistics). The set of all possible decision outcomes constitutes of a decision space, denoted by $\mathfrak{D}$. For the requirements of the measure theoretical developments, a $\sigma$-algebra on $\mathfrak{D}$ is necessary, denoted by $\mathfrak{B}(\mathfrak{D})$. Thus $(\mathfrak{D}, \mathfrak{B}(\mathfrak{D}))$ is called as a decision space.

(3) **Loss function**. Whether a decision is bad or good, there must be a platform for comparison. For given probability distributional family $\{(\mathfrak{X}, \mathfrak{B}_{\mathfrak{X}}, P_\theta), \theta \in \Theta\}$ and the decision space $\mathfrak{D}$, we introduce a loss function associated with a specific decision $d$, $d \in \mathfrak{D}$, for the justification of decision merit.

**Definition 14:** Let the parameter space be $\Theta$ and the decision space is $(\mathfrak{D}, \mathfrak{B}(\mathfrak{D}))$. Any function defined on the product space $\Theta \times \mathfrak{D}$ is a loss function, denoted as $L(\theta, d)$, if it is satisfies the following two conditions:

(a) $0 \le L(\theta, d) < \infty$ for any $\theta \in \Theta$ and any $d \in \mathfrak{D}$;

(b) For any fixed $\theta \in \Theta$, $L(\theta, d)$ as function of $d$ is $\mathfrak{B}(\mathfrak{D})$-measurable.

The specification of loss sets up the criterion for decision choices, namely control of loss.

### 3.5.4.2 Decision function

Having the descriptions of the three basic elements of statistical decision problem, we note that for any concrete problem, decision making aims to select a good decision $d$ in $\mathfrak{D}$, which depends upon the value of loss function $L(\theta,d)$. If $\theta$ is given, the problem is easily settled. Given a value of $\theta$, the distribution of $X$, $P_\theta$, is known. If $\theta$ is not given, it is necessary to utilize the information of the observational data and the underlying distribution of $X$ contained in the sample values $x$ (to infer the value $\theta$) in order to support the decision maker to make the choice.

Therefore, the task of a statistical decision is just to establish a function $\delta(x)$ called a statistical decision function, defined on the sample space $(\mathfrak{X},\mathfrak{B}(\mathfrak{X}))$ and taking values on decision space $(\mathfrak{D},\mathfrak{B}(\mathfrak{D}))$ such that when a sample value $x$ is available, the value of the decision $\delta(x)$ will be determined.

**Definition 15:** (Non-randomized statistical decision function) Let the sample space be $(\mathfrak{X},\mathfrak{B}(\mathfrak{X}))$ and the decision space be $(\mathfrak{D},\mathfrak{B}(\mathfrak{D}))$. Any measurable transformation $\delta(x)$ defined on $\mathfrak{X}$ and taking values on $\mathfrak{D}$ is termed a non-randomized statistical decision function.

For any individual decision problem, there are many possible decision functions available. It is necessary to introduce a numerical index, the risk function, to reflect the quality of decision function $\delta(x)$.

**Definition 16:** (Risk function) Suppose that the sample space and distributional family is given by $(\mathfrak{X},\mathfrak{B}(\mathfrak{X}),P_\theta)$, $\theta\in\Theta$, the decision space is $(\mathfrak{D},\mathfrak{B}(\mathfrak{D}))$, the loss function is $L(\theta,d)$, and $\delta(x)$ is a decision function. A function of $\theta$, called the risk function denoted as $R(\theta,\delta)$ is defined with respect to a decision function $\delta$, as

$$\begin{aligned} R(\theta,\delta) &= \mathrm{E}_\theta\left[L(\theta,\delta(X))\right] \\ &= \int_{\mathfrak{X}} L(\theta,\delta(x))\,dP_\theta(x), \quad \theta\in\Theta \end{aligned} \tag{14}$$

In other words, a risk function is the average loss if the decision $\delta(x)$ is taken for whatever the random value $x$ is observed, when the true parameter $\theta$ is given (or assumed). It is obvious that the lesser the risk, the better the decision.

**Definition 17:** (Randomized statistical decision function) Let the sample space be $(\mathfrak{X},\mathfrak{B}(\mathfrak{X}))$ and the decision space be $(\mathfrak{D},\mathfrak{B}(\mathfrak{D}))$. A function defined on the space $\mathfrak{X}\times\mathfrak{B}(\mathfrak{D})$, $\delta(x,D)$, is a randomized statistical decision function, if

(a) for any fixed $D\in\mathfrak{B}_{\mathfrak{D}}$, $\delta(x,D)$ as the function of $x$ is $\mathfrak{B}(\mathfrak{X})$-measurable;

(b) for any fixed $x\in\mathfrak{X}$, $\delta(x,D)$ as the function of D is a probability measure on $\mathfrak{B}(\mathfrak{D})$.

In adopting a randomized statistical decision function $\delta$, the procedure for obtaining a decision d is as follows: first, obtain a random sample $x$ by observing the population $X$, then in terms of $\delta$, obtain the probability measure $\delta(x,D)$ on $\mathfrak{B}(\mathfrak{D})$, and finally, in terms of the probability measure $\delta(x,D)$, select a decision d from the decision space $\mathfrak{D}$. It is obvious that the previously defined non randomized statistical decision function is a special case of the randomized statistical decision function defined here, i.e., for any $x \in \mathfrak{X}$, probability distribution $\delta(x,D)$ is concentrated at a point in D as the function of D (Clearly this point is partially determined by x, i.e., $\delta(x)$ in Definition 15).

One might query the significance of introducing such a very abstract and seemingly unnatural concept. At this stage, we simply stress that the concept will bring certain theoretical conveniences.

The risk function of a randomized statistical decision function is described as follows. Let the sample space and the distributional family be $\{(\mathfrak{X},\mathfrak{B}(\mathfrak{X}),P_\theta),\theta\in\Theta\}$, decision space be $(\mathfrak{D},\mathfrak{B}(\mathfrak{D}))$, and the loss function be $L(\theta,d)$. In order to calculate the risk function $R(\theta,\delta)$ for the randomized statistical decision function we set $\delta=\delta(x,D)$. It is easy to establish that given the sample $x$, the conditional risk of $\delta$ is

$$R(\theta,\delta\,|\,x)=\int_{\mathfrak{X}} L(\theta,\omega)\delta(x,d\omega) \tag{15}$$

where $\omega$ is the moving point in $\mathfrak{D}$. It is necessary to note that

$$\begin{aligned} R(\theta,\delta\,|\,x)&=\int_{\mathfrak{X}} L(\theta,\omega)\delta(x,d\omega) \\ &=\int_{\mathfrak{X}} dP_\theta\left[\int_{\mathfrak{D}} L(\theta,\omega)\delta(x,d\omega)\right] \end{aligned} \tag{16}$$

In order to make the definition effective, it is necessary to verify: the integrals in (2) and (3) are meaningful. Note that the functions involved are all nonnegative. Hence once the measurability of the functions is confirmed, the conclusion is reached. $L(\theta,\omega)$ as a function of $\omega$ is $\mathfrak{B}(\mathfrak{D})$-measurable. Now assuming the integral in (2) to be meaningful, then the integral in (3) must be meaningful. This confirmation arises from showing that $R(\theta,\delta\,|\,x)$ as a function of $x$, is $\mathfrak{B}(\mathfrak{X})$ -measurable. In other words, for any nonnegative $\mathfrak{B}(\mathfrak{D})$-measurable function $f(\omega)$, the function

$$\gamma(x)=\int_{\mathfrak{D}} f(\omega)\delta(x,d\omega) \tag{17}$$

must be $\mathfrak{B}(\mathfrak{X})$-measurable. This property can be shown by the standard measure theoretical approach: start with indicator function $f(\omega)=\vartheta_D(\omega)$, $D\in\mathfrak{B}(\mathfrak{D})$, then a simple function $f(\omega)=\sum_{i=1}^{n}\vartheta_{D_i}(\omega),\ D_i\in\mathfrak{B}(\mathfrak{D})$, and finally an arbitrary nonnegative function $f(\omega)$, which can be approached by limiting process by specifying a sequence of nonnegative simple functions $\{f_n(\omega)\}$.

### 3.5.5 Elements of uncertainty decision problem



Now, with the descriptions of the statistical decision theory, we may attempt to discuss the construction of a general uncertainty decision theory in a comparable manner.

It is necessary to point out the three components, i.e., the state, action, and loss in the statistical decision theory, are still the essential elements in the new general uncertainty decision theory. (Note, the term for the first element, "state" is used here). However, the connotations inherent in the three elements are not always the same as in the statistical approach. Let us examine element by element.

Firstly, in statistical decision theory, the state, termed "state of nature" (i.e., sample space and distributional family), is regarded as objectively in existence, at least in some consensus sense, while in any general uncertainty environments, the state may include subjective judgmental or even phenomenological events or factors. For example, top decision makers may include company's middle managements' or engineers possible cover-up behaviour as one of the "state" elements, which need not be observable and non-repeatable events. (Such possible information may sound dirty, spurious or problematic, and the decision makers might never wish to release this approach to the employees or the public). Note here the conceptual interpretations of state acquire when involving the decision environments, i.e., "reality" ahead of the decision makers, possible virtual actions, and virtual loss. The differentiation between the "state of nature" in the statistical decision theory and the "state" in the uncertainty decision theory is critical. The former is under the frequentist statistician's "reality" inferences, more or less reflecting the "truth", while the latter is a mixture of subjective and objective reflections.

Secondly, the connotations of "action" in the statistical decision theory and that in the uncertainty decision theory can be discerned. The action in the statistical decision theory is the possible reaction and treatment against the state event. The state element is the root cause and accordingly, the action is selected against the cause. It is an if-then logic and thus a many to one mapping from the state space onto action space (point-wise).

While the “action” in the uncertainty decision theory is not necessarily a one-to-one mapping consequence because particular “states” under consideration may be of artificial, or personal experience, or a phenomenological nature. A director with rich experience may not be afraid in facing a crisis and they may even delicately utilize the crisis to create more development chances instead. The state space in an experienced director’s mind is likely to be much enlarged by inclusion of many “states” due to specific understanding of crisis, in contrast with an inexperienced junior manager, Therefore, the action space is virtual, in which some elements are of a precautionary nature and do not correspond to any specific state element. The mapping is of multiple states to multiple action nature. However, the inclusion of virtual action elements is extremely important, because the top decision maker does not need to deal with routine decisions of day-to-day operations but with the extreme event or the most important event decision.

Thirdly, the loss in both decision theories is the same. However, the social loss and environmental loss occupy more and more concern from the public, NGO and the governmental agencies as well. In the new uncertainty decision theory, safety factor state, health factor state, and environmental factor state should be automatically assigned uncertain measure grades because of their intrinsic features.

According to statistical decision theory, the decision is made in terms of observational data, denoted as $z$, which is described by an probability distribution $F(z;\theta)$. Based on data $z$ (i.e., $F(z;\theta)$), a decision is actually a mapping from data space $\mathbb{D}$ into action space $\mathbb{A}$. In other words,

$$a:\mathbb{D}\rightarrow\mathbb{A} \tag{18}$$

which can be expressed by

$$a=d(z) \tag{19}$$

The action taken is random because the observational data to which decision rule is applied are random. Consequently, the loss function $l(\theta,d(z))$ is random in statistical decision theory. Therefore, $\{(\mathfrak{X},\mathfrak{B}(\mathfrak{X}),P_\theta),\theta\in\Theta\}$, the probability distribution family is constituted of the elementary mechanism underlying the statistical decision processes.

Similarly, in section two, where the uncertainty distribution theory is introduced, we stress that $\{(\mathbb{Z},\mathfrak{B}(\mathbb{Z}),\Psi_\tau),\tau\in\Gamma\}$, the uncertainty distribution family is constituted of the fundamental mechanism underlying the uncertainty decision processes. Let us explore what an uncertainty distribution may look like via a detailed example.

**Example 18:** Let $\xi$ be an uncertain variable, which takes values on $\mathbb{Z}=\{0,0+,1,1+,2,2+,3,3+,4\}$ with the uncertainty distribution:

$$\Psi_\xi(z)=\begin{cases} 0 \quad z<0 \\ 0.25 \quad z=0 \\ 0.2z+0.25 \quad 0<z<1 \\ 0.575 \quad z=1 \\ 0.125z+0.45 \quad 1<z<2 \\ 0.77 \quad z=2 \\ 0.07z+0.63 \quad 2<z<3 \\ 0.85 \quad z=3 \\ 0.09z+0.59 \quad 3<z<4 \\ 1.0 \quad z\ge 4 \end{cases} \tag{20}$$

which implies

$$\begin{aligned} \pi_0 &= \lambda\{\xi=0\}=0.25, \\ \pi_{0+} &= \lambda\{\xi=0+\}=0.05, \\ \pi_{1-} &= \lambda\{\xi=1-0\}=0.15, \\ \pi_1 &= \lambda\{\xi=1\}=0.125, \\ \pi_{1+} &= \lambda\{\xi=1+0\}=0.015, \\ \pi_{2-} &= \lambda\{\xi=2-0\}=0.11, \\ \pi_2 &= \lambda\{\xi=2\}=0.07, \\ \pi_{2+} &= \lambda\{\xi=2+0\}=0.03, \\ \pi_{3-} &= \lambda\{\xi=3-0\}=0.04, \\ \pi_3 &= \lambda\{\xi=3\}=0.01, \\ \pi_{3+} &= \lambda\{\xi=3+0\}=0.08, \\ \pi_{4-} &= \lambda\{\xi=4-0\}=0.02, \\ \pi_4 &= \lambda\{\xi=4-0\}=0.05 \end{aligned} \tag{21}$$

**Remark 19:** It is easily seen that the uncertain distribution $\Psi$ defined in Example 18 is a function with finite jumps, where $\lim_{z\uparrow z_i}\xi=c_{i-}<c_{i+}=\lim_{z\downarrow z_i}\xi$ and $\xi=z_i=c_i$, $c_{i-}<c_i<c_{i+}$, satisfying Theorem 7, although $\{z=c_i\}$, $i=0,1,\cdots,4$ are so-called removable points in calculus theory. It can

be further verified that the uncertain distribution $\Psi$ is neither left continuous nor right-continuous. $\Psi$ in this example gives an elementary form of an uncertain distribution.

**Definition 20:** (Essential Form of an Uncertain Distribution) Let $\xi$ be an uncertain variable with essential form, which takes its values from an ascending ordered domain set $\mathbb{D}=\{c_0, c_1, \cdots, c_n\}$ with the uncertain distribution $\Psi$ defined by

$$\begin{aligned}
&\Psi(c_0) = \bar{\lambda}\{\xi \le c_0\} = 0, \\
&\Psi(c_1 -) = \psi_{1-},\ \Psi(c_1) = \psi_1,\ \Psi(c_1 +) = \psi_{1+}, \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots, \\
&\Psi(c_i -) = \psi_{i-},\ \Psi(c_i) = \psi_i,\ \Psi(c_i +) = \psi_{i+}, \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots, \\
&\Psi(c_n -) = \psi_{n-}, \Psi(c_n) = 1.00
\end{aligned} \tag{22}$$

such that $\psi_{i-} < \psi_i < \psi_{i+}$, $i = 1, 2, \cdots, n$. Furthermore, it requires $\pi_i = \bar{\lambda}\{\xi = c_i\} \in (0, \psi_{i+} - \psi_{i-})$, $i = 1, 2, \cdots, n$.

**Theorem 21:** Let $\xi$ be an uncertain variable with the essential form, which takes values from ascending ordered domain set $\mathbb{D}=\{c_0, c_1, \cdots, c_n\}$ with the uncertain distribution. Then $\Psi$ satisfies the following necessary and sufficient conditions:

(i)

$$\Psi(c_0) = \bar{\lambda}\{\xi \le c_0\} = 0 \tag{23}$$

(ii) For $i = 1, 2, \cdots, n-1$,

$$\begin{aligned}
&\Psi(c_i -) = \bar{\lambda}\{\xi < c_i\} = \psi_{i-}, \\
&\Psi(\{c_i\}) = \bar{\lambda}\{\xi = c_i\} = \pi_i, \\
&\bar{\Psi}(c_i +) = \bar{\lambda}\{\xi > c_i\} = \psi_{i+}
\end{aligned} \tag{24}$$

(iii)

$$\begin{aligned}
&\Psi(c_n -) = \bar{\lambda}\{\xi < c_n\} = \psi_{n-}, \\
&\Psi(\{c_n\}) = \bar{\lambda}\{\xi = c_n\} = \pi_n, \\
&\Psi(c_n) = \bar{\lambda}\{\xi \le c_n\} = 1.00
\end{aligned} \tag{25}$$

(iv) The uncertain measure of singleton $\{c_i\}$

$$\begin{aligned}\pi_{i-} &= \lambda\{\xi = c_i - 0\},\\ \pi_i &= \lambda\{\xi = c_i\},\\ \pi_{i+} &= \lambda\{\xi = c_i + 0\},\end{aligned} \tag{26}$$

such that $\sum_{i=0}^{n}\left(\pi_{i-} + \pi_i + \pi_{i+}\right) = 1$

**Definition 22:** If an uncertainty distribution takes the form

$$\Psi_d(z) = \begin{cases} 0 & z < c_0 \\ \pi_1 & z = c_1 \\ \pi_{1+} & c_1 < z < c_2 \\ \pi_2 & z = c_2 \\ \vdots & \vdots \\ \pi_i & z = c_i \\ \pi_{i+} & c_i < z < c_{i+1} \\ \pi_{i+1} & z = c_{i+1} \\ \vdots & \vdots \\ 1.0 & z \geq c_m \end{cases} \tag{27}$$

where $0 < \pi_i < \pi_{i+} < \pi_{i+1} < 1$, then it is a discrete uncertain distribution.

**Theorem 23:** The expectation of a discrete uncertainty distribution $\Psi_d$, denoted as $\mathrm{E}_\Psi[\xi]$, is given by

$$\mathrm{E}_\Psi[\xi] = \sum_{i=0}^{n} w_i c_i \tag{28}$$

where

$$\begin{aligned} w_i = &\max_{0\leq j\leq n}\left\{\pi_j, \pi_{j+} \mid c_j \leq c_i\right\} \wedge 0.5 \\ &- \max_{0\leq j\leq n}\left\{\pi_j, \pi_{j+} \mid c_j < c_i\right\} \wedge 0.5 \\ &+ \max_{0\leq j\leq n}\left\{\pi_j, \pi_{j+} \mid c_j \geq c_i\right\} \wedge 0.5 \\ &- \max_{0\leq j\leq n}\left\{\pi_j, \pi_{j+} \mid c_j > c_i\right\} \wedge 0.5 \end{aligned} \tag{29}$$

$i = 0, 1, 2, \cdots, m$.

**Proof:** The proof of Theorem 23 is just the application of Liu's (2010) definition of uncertain expectation to a discrete uncertain variable with neither left-continuity nor right-continuity:

$$\mathrm{E}[\xi] = \int_0^{+\infty} \lambda\{\xi \geq s\}\, ds - \int_{-\infty}^{0} \lambda\{\xi \leq s\}\, ds \tag{30}$$

**Example 24:** Calculate the expectation of the discrete uncertain variable defined by

$$\Psi_\xi(z) = \begin{cases} 0 & z < 0 \\ 0.25 & z = 0 \\ 0.45 & 0 < z < 1 \\ 0.575 & z = 1 \\ 0.7 & 1 < z < 2 \\ 0.77 & z = 2 \\ 0.84 & 2 < z < 3 \\ 0.85 & z = 3 \\ 0.95 & 3 < z < 4 \\ 1.0 & z \geq 4 \end{cases} \tag{31}$$

Let us calculate the weight $w_i$, $i = 0,1,2,3,4$ . Note that the uncertain measure grades can be written

$$\begin{aligned} \pi_0 &= \lambda\{\xi = 0\} = 0.25, \\ \pi_{0+} &= \lambda\{\xi = 0+\} = 0.20, \\ \pi_1 &= \lambda\{\xi = 1\} = 0.125, \\ \pi_{1+} &= \lambda\{\xi = 1+0\} = 0.125, \\ \pi_2 &= \lambda\{\xi = 2\} = 0.07, \\ \pi_{2+} &= \lambda\{\xi = 2+0\} = 0.07, \\ \pi_3 &= \lambda\{\xi = 3\} = 0.01, \\ \pi_{3+} &= \lambda\{\xi = 3+\} = 0.10, \\ \pi_4 &= \lambda\{\xi = 4-0\} = 0.05 \end{aligned} \tag{32}$$

Then we can calculate the weights:

$$\begin{aligned} w_0 &= \max_{0\leq j\leq n}\{\pi_j \mid j \leq 0\} \wedge 0.5 - \max_{1\leq j\leq n}\{\pi_j \mid j < 0\} \wedge 0.5 \\ &\quad + \max_{1\leq j\leq n}\{\pi_{j+}, \pi_j \mid j \geq 0\} \wedge 0.5 - \max_{1\leq j\leq n}\{\pi_{j+}, \pi_j \mid j > 0\} \wedge 0.5 \\ &= 0.25 - 0.00 + 0.20 - 0.125 = 0.325 \\ w_1 &= \max_{0\leq j\leq n}\{\pi_j, \pi_{j+} \mid j \leq 1\} \wedge 0.5 - \max_{0\leq j\leq n}\{\pi_j, \pi_{j+} \mid j < 1\} \wedge 0.5 \\ &\quad + \max_{0\leq j\leq n}\{\pi_j, \pi_{j+} \mid j \geq 1\} \wedge 0.5 - \max_{0\leq j\leq n}\{\pi_j, \pi_{j+} \mid j > 1\} \wedge 0.5 \\ &= 0.25 - 0.20 + 0.125 - 0.125 = 0.05 \end{aligned} \tag{33}$$

$$
\begin{aligned}
w_2 &= \max_{0\le j\le n}\left\{\pi_{j-},\pi_j,\pi_{j+} \mid j\le 2\right\}\wedge 0.5-\max_{0\le j\le n}\left\{\pi_{j-},\pi_j,\pi_{j+} \mid j< 2\right\}\wedge 0.5\\
&\quad+\max_{0\le j\le n}\left\{\pi_{j-},\pi_j,\pi_{j+} \mid j\ge 2\right\}\wedge 0.5-\max_{0\le j\le n}\left\{\pi_{j-},\pi_j,\pi_{j+} \mid j> 2\right\}\wedge 0.5\\
&= 0.25-0.25+0.10-0.10=0.00
\end{aligned}
$$

$$
\begin{aligned}
w_3 &= \max_{0\le j\le n}\left\{\pi_{j-},\pi_j,\pi_{j+} \mid j\le 3\right\}\wedge 0.5-\max_{0\le j\le n}\left\{\pi_{j-},\pi_j,\pi_{j+} \mid j< 3\right\}\wedge 0.5\\
&\quad+\max_{0\le j\le n}\left\{\pi_{j-},\pi_j,\pi_{j+} \mid j\ge 3\right\}\wedge 0.5-\max_{0\le j\le n}\left\{\pi_{j-},\pi_j,\pi_{j+} \mid j> 3\right\}\wedge 0.5\\
&= 0.25-0.25+0.10-0.10=0.00
\end{aligned}
$$

and

$$
\begin{aligned}
w_4 &= \max_{0\le j\le n}\left\{\pi_{j-},\pi_j,\pi_{j+} \mid j\le 4\right\}\wedge 0.5-\max_{0\le j\le n}\left\{\pi_{j-},\pi_j,\pi_{j+} \mid j< 4\right\}\wedge 0.5\\
&\quad+\max_{0\le j\le n}\left\{\pi_{j-},\pi_j,\pi_{j+} \mid j\ge 4\right\}\wedge 0.5-\max_{0\le j\le n}\left\{\pi_{j-},\pi_j,\pi_{j+} \mid j> 4\right\}\wedge 0.5\\
&= 0.25-0.25+0.05-0.00=0.05
\end{aligned}
\tag{34}
$$

Hence the expected value of the uncertain variable is

$$
\begin{aligned}
\mathrm{E}\left[\xi\right] &= 0.25\times 0+0.05\times 1+0.00\times 2\\
&\quad+0.00\times 3+0.05\times 4=0.25
\end{aligned}
\tag{35}
$$

**Remark 25:** Recall that in statistical decision, the index, risk, is an expectation involving a distribution function. We cannot compare two random quantities directly because, until the statistical experiment is finished and the outcomes are obtained, we do not know their values. Furthermore, because each individual outcome (realization) of a random variable (quantity) is associated with a probability, simply comparing the values of two random quantities will let decision-making ignore their associated probability grades. Therefore, statistical decision is commonly made in a weighted average sense, i.e., in expectation. The risk concept simply reflects the expectation feature. Similarly, it is fair to say the uncertainty decision is also made using expectation of an uncertainty distribution.

The fact that the Toyota management was aware of faulty car component problems, but did not address potential impacts of social risk and environmental awareness, may reveal that their description of the state space is questionable. In other words, the state space $\Theta$ used in the Toyota decision might include the faulty parts sub-space $\Theta_f \subset \Theta$, but the associated measure grades assigned to the states in $\Theta_f$were so tiny so that the decision function specification essentially generates near-null loss, and hence decisions unaffected by the parts are preferable to the management's thinking.

According to Liu (2010), the uncertain distribution function on the state space is a prior distribution, $p(\Theta)$, which may or may not be updateable by the information in data form. Bayes theorem (in probability) may also provide a updating structure for the state distribution, whose result is the called posterior uncertain distribution on the state $p(\Theta|z)$.

### 3.5.6 A Discrete Uncertainty Decision

**Definition 26:** An uncertain decision is a selection, which minimizes the loss function $l(\theta,a)$ or regret function $r(\theta,a)=l(\theta,a)-\min_{a\in\mathbb{A}}\{l(\theta,a)\}$ of an action $a$ from action space $\mathbb{A}$ for given state $\theta$ in the state space $\Theta$.

**Definition 27:** The expected value of the loss with respect to the distribution of uncertain data $z$.

$$R\ \theta,d\ =\mathrm{E}_\theta\left[l\ \theta,d\ z\ \right] \tag{36}$$

is called the risk function

**Remark 28:** The distribution of uncertain data $z$ depends on state $\theta$, because the dependence of $R(\theta,d)$ on $\theta$ enters explicitly from $l(\theta,a)$and also through the state $\theta$ in the distribution function $\Psi(z,\theta)$ for $z$.

**Example 29:** Two quality states $\Theta=\{\theta_1,\theta_2\}$, where $\theta_1$: Liu's [15] uncertain quality state, $\theta_2$, Gaussian quality state are assumed.

The uncertain variable $\xi$ is discrete variable taking values on $\mathbb{Z}=\{0,0+,1,1+,2,2+,3,3+,4\}$.

Then $\Psi(z;\theta)$, the uncertain distribution given $\theta_1$ is:

$$\Psi_\xi(z|\theta_1)=\begin{cases} 0 & z<0 \\ 0.71 & z=0 \\ 0.71964 & 0<z<1 \\ 0.93964 & z=1 \\ 0.94822 & 1<z<2 \\ 0.99122 & z=2 \\ 0.99138 & 2<z<3 \\ 0.99938 & z=3 \\ 0.99998 & 3<z<4 \\ 1.0 & z\geq 4 \end{cases} \tag{37}$$

Note that the uncertain measure grades given $\theta_1$ are

$$
\begin{aligned}
\pi_0 &= \lambda\{\xi = 0\} = 0.71, \\
\pi_{0+} &= \lambda\{\xi = 0+\} = 0.00964, \\
\pi_1 &= \lambda\{\xi = 1\} = 0.22, \\
\pi_{1+} &= \lambda\{\xi = 1+0\} = 0.00858, \\
\pi_2 &= \lambda\{\xi = 2\} = 0.04, \\
\pi_{2+} &= \lambda\{\xi = 2+0\} = 0.0032, \\
\pi_3 &= \lambda\{\xi = 3\} = 0.008, \\
\pi_{3+} &= \lambda\{\xi = 3+\} = 0.0006, \\
\pi_4 &= \lambda\{\xi = 4\} = 0.00002
\end{aligned}
\tag{38}
$$

The uncertain distribution $\Psi(z;\theta)$ given $\theta_2$ is:

$$
\Psi_\xi(z \mid \theta_2) = \begin{cases}
0 & z < 0 \\
0.68 & z = 0 \\
0.68268 & 0 < z < 1 \\
0.95268 & z = 1 \\
0.9545 & 1 < z < 2 \\
0.9965 & z = 2 \\
0.9973 & 2 < z < 3 \\
0.9993 & z = 3 \\
0.99995 & 3 < z < 4 \\
1.0 & z \geq 4
\end{cases}
\tag{39}
$$

Note that the uncertain measure grades are

$$
\begin{aligned}
\pi_0 &= \lambda\{\xi = 0\} = 0.68, \\
\pi_{0+} &= \lambda\{\xi = 0+\} = 0.00268, \\
\pi_1 &= \lambda\{\xi = 1\} = 0.27, \\
\pi_{1+} &= \lambda\{\xi = 1+0\} = 0.00182, \\
\pi_2 &= \lambda\{\xi = 2\} = 0.042, \\
\pi_{2+} &= \lambda\{\xi = 2+0\} = 0.0008, \\
\pi_3 &= \lambda\{\xi = 3\} = 0.002, \\
\pi_{3+} &= \lambda\{\xi = 3+\} = 0.00065, \\
\pi_4 &= \lambda\{\xi = 4\} = 0.00005
\end{aligned}
\tag{40}
$$

The loss is defined in the Table 3.5.1. The uncertain distribution is defined by Table 3.5.2.

Table 3.5.1 A loss function in tabular form

| $l(\theta,a)$ | $\theta_1$ | $\theta_2$ |
|---|---|---|
| $a_1$ | 1.2 | 4.5 |
| $a_2$ | 3.5 | 1.5 |

Table 3.5.2 Data uncertain distribution

| $\Pr\{Z=z_i \mid \theta_j\}$ | $\theta_1$ | $\theta_2$ |
|---|---|---|
| $\xi=z_1=0$ | 0.71 | 0.68 |
| $\xi=z_2=0+$ | 0.00964 | 0.00268 |
| $\xi=z_3=1$ | 0.22 | 0.27 |
| $\xi=z_4=1+$ | 0.00858 | 0.00182 |
| $\xi=z_5=2$ | 0.004 | 0.042 |
| $\xi=z_6=2+$ | 0.0032 | 0.0008 |
| $\xi=z_7=3$ | 0.008 | 0.002 |
| $\xi=z_8=3+$ | 0.0006 | 0.00065 |
| $\xi=z_9=4$ | 0.00002 | 0.00005 |

Then the decision function will have $9\times 2=18$ elements since there are 2 actions and 9 observations:

Table 3.5.3 Uncertain decision function $a=d(z)$

| $d(z)$ | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ | $d_6$ | $d_7$ | $d_8$ | $d_9$ |
|---|---|---|---|---|---|---|---|---|---|
| $z_1$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_2$ |
| $z_2$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_2$ | $a_2$ |
| $z_3$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_2$ | $a_2$ | $a_2$ |
| $z_4$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ |
| $z_5$ | $a_1$ | $a_1$ | $a_1$ | $a_1$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ |
| $z_6$ | $a_1$ | $a_1$ | $a_1$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ |
| $z_7$ | $a_1$ | $a_1$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ |
| $z_8$ | $a_1$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ |
| $z_9$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ | $a_2$ |

where the decision space $\mathbb{D}=\{d_1,d_2,\cdots,d_9\}$.

Table 3.5.4 Risk function $R(\theta,d) = \mathrm{E}_\theta\left[l(\theta,d(z))\right]$

| $R(\theta,d)$ | $\theta_1$ | $\theta_2$ |
|---|---|---|
| $d_1$ | 1.200066 | 3.4999 |
| $d_2$ | 1.202046 | 3.4986 |
| $(\mathfrak{D},\mathfrak{B}(\mathfrak{D}))$ | 1.228446 | 3.4946 |
| $d_4$ | 1.228974 | 3.493 |
| $d_5$ | 1.370874 | 3.40885 |
| $d_6$ | 1.68108 | 3.40536 |
| $d_7$ | 2.125188 | 2.88036 |
| $d_8$ | 2.157 | 2.86 |
| $d_9$ | 4.5 | 1.5 |

To demonstrate the calculation of entries in Table 3.5.4, we evaluate $R(\theta_i,d_j),i=1,2; j=1,2,\cdots,9$,

$$
\begin{aligned}
&R(\theta_1,d_1) = \\
&= l(\theta_1,a_1)p(z=z_1\,|\,\theta_1) + l(\theta_1,a_1)p(z=z_2\,|\,\theta_1) \\
&+ l(\theta_1,a_1)p(z=z_3\,|\,\theta_1) + l(\theta_1,a_1)p(z=z_4\,|\,\theta_1) \\
&+ l(\theta_1,a_1)p(z=z_5\,|\,\theta_1) + l(\theta_1,a_1)p(z=z_6\,|\,\theta_1) \\
&+ l(\theta_1,a_1)p(z=z_7\,|\,\theta_1) + l(\theta_1,a_1)p(z=z_8\,|\,\theta_1) \\
&+ l(\theta_1,a_2)p(z=z_9\,|\,\theta_1) \\
&= 1\times 0.99998 + 4\times 0.00002 = 1.00006 \\
&R(\theta_1,d_2) \\
&= l(\theta_1,a_1)p(z=z_1\,|\,\theta_1) + l(\theta_1,a_1)p(z=z_2\,|\,\theta_1) \\
&+ l(\theta_1,a_1)p(z=z_3\,|\,\theta_1) + l(\theta_1,a_1)p(z=z_4\,|\,\theta_1) \\
&+ l(\theta_1,a_1)p(z=z_5\,|\,\theta_1) + l(\theta_1,a_1)p(z=z_6\,|\,\theta_1) \\
&+ l(\theta_1,a_1)p(z=z_7\,|\,\theta_1) + l(\theta_1,a_2)p(z=z_8\,|\,\theta_1) \\
&+ l(\theta_1,a_2)p(z=z_9\,|\,\theta_1) \\
&= 1\times 0.99938 + 4\times 0.00062 = 1.00186
\end{aligned} \tag{41}
$$

With the specifications of risk function, a further criterion for selecting a decision must be defined. As an example, we employ the minimax principle. The first step is to maximise the risk with respect to state space, i.e.,

$$d^M = \max_\theta \{R(\theta,d)\} \tag{42}$$

And the second step is to find the minimax decision rule which minimizes $d^M$

$$d^{mM} = \min_d \left\{ d^M \right\} = \min_d \left\{ \max_\theta \left\{ R(\theta, d) \right\} \right\} \tag{43}$$

**Example 30:** Minimax decision rule for Example 29.

Table 3.5.5 Minimax decision rule $d^{mM}$ search

| | $\theta_1$ | $\theta_2$ | $M(d)$ |
|---|---|---|---|
| $d_1$ | 1.200066 | 3.4999 | 3.4999 |
| $d_2$ | 1.202046 | 3.4986 | 3.4986 |
| $d_3$ | 1.228446 | 3.4946 | 3.4946 |
| $d_4$ | 1.228974 | 3.493 | 3.493 |
| $d_5$ | 1.370874 | 3.40885 | 3.40885 |
| $d_6$ | 1.68108 | 3.40536 | 3.40536 |
| $d_7$ | 2.125188 | 2.88036 | 2.88036 |
| $d_8$ | 2.157 | 2.86 | 2.86 |
| $d_9$ | 4.5 | 1.5 | 4.5 |
| $d^{mM}$ | | | 2.86 |

The development of Section 3.5.6 reveals that in a discrete uncertainty decision, the procedure for the uncertainty decision is similar to that for statistical decision. The fundamental difference lies in the connotation of the "state" and underlying uncertainty distribution as discussed in Section 3.5.5.



### 3.5.7 A continuous uncertain decision

In Section 3.5.6 we explored the uncertain decision given discrete state space, action space and discrete loss function environments. Now, we investigate the decision problem under a continuous uncertainty environment.

Recall that Definition 20 states the essential form of the uncertainty distribution. It is neither left-continuous nor right-continuous and its distribution function has finite jumps and "removable" values at jump points. If the distribution function is continuous everywhere, i.e., there is no jump and no removable point in its domain, it is a continuous uncertainty distribution.

Peng and Iwamura (2009) give an uncertain variable $\xi(x) = x$ defined on the uncertain space $(\Xi, \mathfrak{A}, \lambda)$ where

$$\lambda\{A\} = \begin{cases} 0 & \text{if } A = \varnothing \\ c & \text{if } A \text{ is upper bounded} \\ 0.5 & \text{if } A \text{ and } A^c \text{ are both upper bounded} \\ 1-c & \text{if } A^c \text{ is upper bounded} \\ 1 & \text{if } A = \mathbb{R} \end{cases} \tag{44}$$

Then the uncertain distribution for $\xi$ is $\Psi(x) = c,\ 0 < c < 0.5$.

Another continuous uncertainty distribution example is Liu's [15] uncertain normal distribution

$$\Psi(z; e, \sigma) = \frac{1}{1 + e^{-\frac{\pi}{\sqrt{3}\sigma}(z-e)}}, \quad z \in \mathbb{R} \tag{45}$$

Let us consider the uncertain decision problem for a given continuous distribution. Assume state space $\Theta = \mathbb{R}$, and action space $\mathbb{A} = \mathbb{R}$, and the loss function defined by

$$l(\theta, a) = w(\theta)(\theta - a)^2 \tag{46}$$

i.e., a quadratic loss function is assumed.

**Definition 31:** (Uncertain Bayes loss) Given a continuous state space $\Theta$, the uncertain variable $\theta$ is defined on uncertain space $(\Theta, \mathfrak{B}(\Theta), \lambda)$, where $\lambda(\cdot)$ is an uncertain measure. The uncertain distribution $\Psi(\theta)$ is defined on $(\Theta, \mathfrak{B}(\Theta))$ ). Then we seek the average of loss with respect to state space for a given action $a \in \mathbb{A}$. (Note the action space $\mathbb{A}$ is continuous too.) The quantity

$$B(a) = \mathrm{E}\left[l(\theta, a)\right] = \int_{\Theta} l(\theta, a)\, d\Psi(\theta) \tag{47}$$

is called as the uncertain Bayes loss for a given action $a$.

**Definition 32:** (Uncertain Bayes risk) The Bayes risk is

$$B(d) = \mathrm{E}\left[l(\theta, d(z))\right] = \int_{\Theta} l(\theta, d(z))\, d\Psi(\theta) \tag{48}$$

**Definition 33:** (Uncertain Bayes rule) A Bayes decision rule, denoted as $d^B$ is a rule such that the Bayes risk is minimized, i.e.,

$$B(d^B) = \min_{d \in \mathbb{D}} \{B(d)\} \tag{49}$$

**Example 34:** Given a continuous state space $\Theta = \mathbb{R}$, the uncertain variable $\theta$ is defined on uncertain space $(\mathbb{R}, \mathfrak{B}(\mathbb{R}), \lambda)$, where $\lambda(\cdot)$ is properly defined. The uncertain distribution is

$$\Psi(\theta) = \begin{cases} 0 & \theta \le \alpha \\ \dfrac{\theta - \alpha}{2(\beta - \alpha)} & \alpha < \theta \le \beta \\ \dfrac{\theta + \gamma - 2\beta}{2(\gamma - \beta)} & \beta < \theta \le \gamma \\ 1 & \theta > \gamma \end{cases} \tag{50}$$

Then we find the average of loss with respect to state space for a given action $a \in \mathbb{A}$, as the uncertain Bayes loss:

$$B(a) = \mathrm{E}\left[l(\theta, a)\right] = \int_{\Theta} w(\theta)(\theta - a)^2 \, d\Psi(\theta) \tag{51}$$

Set $w(\theta) = w_0$, a constant. The uncertain Bayes loss is

$$\begin{aligned} B(a) &= \frac{w_0}{2(\beta - \alpha)} \int_{\alpha}^{\beta} (\theta - a)^2 d\theta + \frac{w_0}{2(\gamma - \beta)} \int_{\beta}^{\gamma} (\theta - a)^2 d\theta \\ &= \frac{w_0}{6}\left[(\beta - a)^2 + (\alpha - a)^2 + (\beta - a)(\alpha - a)\right] \\ &+ \frac{w_0}{6}\left[(\gamma - a)^2 + (\beta - a)^2 + (\gamma - a)(\beta - a)\right] \\ &= \frac{w_0}{6}\left(3a^2 - 3(\alpha + \beta)a + \alpha^2 + \beta^2\right) \\ &+ \frac{w_0}{6}\left(3a^2 - 3(\gamma + \beta)a + \gamma^2 + \beta^2\right) \\ &= \frac{w_0}{6}\left(6a^2 - 3(\alpha + 2\beta + \gamma)a + \left(\alpha^2 + 2\beta^2 + \gamma^2\right)\right) \end{aligned} \tag{52}$$

With appropriate specification of decision function in term *of* data, the uncertain Bayesian decision analysis can be formulated.

Liu (2010) states his maximum uncertain principle, (abbreviated as MUP): “for any event, if there are multiple reasonable values that an uncertain measure may take, then the value as close as to 0.5 as possible is assigned to the event”.

**Definition 35:** Let $\omega(\cdot\,|x)$ denote the regular conditional distribution of $\theta$, given $X = x$. If $\{v_\theta, \theta \in \Theta\} \ll \mu$, where $\mu$ is a σ-finite measure on $\mathfrak{B}_{\mathfrak{X}}$, and define $f(\theta, x) = dv_\theta(x) / d\mu$, then for $\forall \mathrm{B} \in \mathfrak{B}_\Theta$,

$$\omega(B \mid x) = \begin{cases} \dfrac{\int_B f(\theta,x)\,d\omega(\theta)}{\int_\Theta f(\theta,x)\,d\omega(\theta)} \vee 0.5 & \text{if } \dfrac{\int_B f(\theta,x)\,d\omega(\theta)}{\int_\Theta f(\theta,x)\,d\omega(\theta)} < 0.5 \\ 1-\dfrac{\int_{B^c} f(\theta,x)\,d\omega(\theta)}{\int_\Theta f(\theta,x)\,d\omega(\theta)} & \text{if } \dfrac{\int_{B^c} f(\theta,x)\,d\omega(\theta)}{\int_\Theta f(\theta,x)\,d\omega(\theta)} < 0.5 \\ 0.5 & \text{othervise} \end{cases} \tag{53}$$

**Theorem 36:** The regular conditional distributional of $\widehat{\Theta}$, after obtaining the sample $x$.

$$\Upsilon_{\widehat{\Theta}}(\theta \mid x) = \omega\left\{\widehat{\Theta} \le \theta \mid x\right\} \tag{54}$$

where $\omega(\cdot \mid x)$ is given by Definition 35.

**Remark 37:** Once the MUP posterior density is specified, the posterior mean and variance can be calculated:

$$\mathrm{E}\left[\widehat{\Theta}\right] = \int_\Theta \theta\, d\Upsilon_{\widehat{\Theta}}(\theta \mid x) \tag{55}$$

and

$$\mathrm{V}\left(\widehat{\Theta}\right) = \mathrm{E}\left[\left(\widehat{\Theta} - \mathrm{E}\left[\widehat{\Theta}\right]\right)^2\right] = \int_\Theta \left(\theta - \mathrm{E}\left[\widehat{\Theta}\right]\right)^2 d\Upsilon_{\widehat{\Theta}}(\theta \mid x) \tag{56}$$

respectively.

**Example 38:** Let the uncertain prior be

$$\omega(\theta) = \begin{cases} 0 & \text{if } \theta \le a \\ \dfrac{\theta - a}{2(b-a)} & \text{if } a < \theta \le b \\ \dfrac{\theta + c - 2b}{2(c-b)} & \text{if } b < \theta \le c \\ 1 & \text{otherwise} \end{cases} \tag{57}$$

hence

$$\frac{d\omega(\theta)}{d\theta} = \begin{cases} \dfrac{1}{2(b-a)} & \text{if } a < \theta \le b \\ \dfrac{1}{2(c-b)} & \text{if } b < \theta \le c \\ 0 & \text{if } \theta \le a \text{ or } \theta > c \end{cases} \tag{58}$$

The sample of size $n$ is drawn *i.i.d.* from a normal family $N(\theta, \sigma_0^2)$, where the variance $\sigma_0^2$ is given. The sample is $x = \{x_1, x_2, \cdots, x_n\}$. Then

$$\frac{dP_\theta}{dx} = \prod_{i=1}^{n} \phi\left(-\frac{x_i - \theta}{\sigma_0}\right) \tag{59}$$

where

$$\phi(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \tag{60}$$

Note that

$$\prod_{i=1}^{n} \phi\left(\frac{x_i - \theta}{\sigma_0}\right) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left(-\frac{(x_i - \theta)^2}{2\sigma_0^2}\right) \tag{61}$$

Since

$$\begin{aligned} \sum_{i=1}^{n} (x_i - \theta)^2 &= n\left(\theta^2 - 2\overline{x}_n \theta\right) + Q^2 \\ &= n\left(\theta - \overline{x}_n\right)^2 + Q^2 - \left(\frac{\overline{x}_n}{2}\right)^2 \end{aligned} \tag{62}$$

where

$$\overline{x}_n = \frac{1}{n} \sum_{i=1}^{n} x_i, \; Q^2 = \sum_{i=1}^{n} x_i^2 \tag{63}$$

Thus

$$\frac{dP_\theta}{dx} = \left(\frac{1}{\sqrt{2\pi}\sigma_0}\right)^n e^{-(\theta - \overline{x}_n)^2 / \left(2\sigma_0^2/n\right)} e^{-\left(Q^2 - (\overline{x}_n/2)^2\right)} \tag{64}$$

The absolute distribution for $\mathbf{x}$ is

$$\begin{aligned} p(x) &= p^*(x) e^{-\left(Q^2 - (\overline{x}_n/2)^2\right)} \\ &= \frac{1}{2(b-a)}\left(\Phi\left(\frac{b - \overline{x}_n}{\sigma_0}\right) - \Phi\left(\frac{a - \overline{x}_n}{\sigma_0}\right)\right) \\ &+ \frac{1}{2(c-b)}\left(\Phi\left(\frac{c - \overline{x}_n}{\sigma_0}\right) - \Phi\left(\frac{b - \overline{x}_n}{\sigma_0}\right)\right) \end{aligned} \tag{65}$$

The posterior density of $\theta$ given $x = \{x_1, x_2, \cdots, x_n\}$ is

$$\gamma(\theta \mid x)=\begin{cases}\dfrac{1}{2(b-a)\,p^{*}(x)}\phi\left(\dfrac{\theta-\bar{x}_n}{\sigma_0/\sqrt{n}}\right) & \text{if } a\le\theta<b\\ \dfrac{1}{2(c-b)\,p^{*}(x)}\phi\left(\dfrac{\theta-\bar{x}_n}{\sigma_0/\sqrt{n}}\right) & \text{if } b\le\theta<c\\ 0 & \text{otherwise}\end{cases} \tag{66}$$

**Example 39:** Suppose that a random sample of size $n$ is taken from the electronic system lifetime with density

$$\frac{dP_\lambda}{dt}=\prod_{i=1}^{n}\lambda e^{-\lambda t_i}=\lambda^n e^{-\lambda T_n} \tag{67}$$

where

$$T_n=\sum_{i=1}^{n}t_i \tag{68}$$

Let us further assume the uncertain prior density

$$\frac{d\omega(\theta)}{d\theta}=\begin{cases}\dfrac{1}{2(b-a)} & \text{if } a<\theta\le b\\ \dfrac{1}{2(c-b)} & \text{if } b<\theta\le c\\ 0 & \text{if } \theta\le a \text{ or } \theta>c\end{cases} \tag{69}$$

Note

$$\begin{aligned}p(t)&=p^{*}(t)\\ &=\frac{1}{2(b-a)}\left(b^n e^{-bT_n}-a^n e^{-aT_n}\right)\\ &+\frac{1}{2(c-b)}\left(c^n e^{-cT_n}-b^n e^{-bT_n}\right)\end{aligned} \tag{70}$$

It is easy to obtain that the posterior density is

$$\omega(\lambda \mid t)=\begin{cases}\dfrac{1}{2(b-a)\,p^{*}(t)}\lambda^n e^{-\lambda T_n} & \text{if } a<\theta\le b\\ \dfrac{1}{2(c-b)\,p^{*}(t)}\lambda^n e^{-\lambda T_n} & \text{if } b<\theta\le c\\ 0 & \text{if } 0\le\theta\le a \text{ or } \theta>c\end{cases} \tag{71}$$

In case of uncertain observations are adequately large, an MUP asymptotic Bayesian analysis can be carried forward immediately by nothing the asymptotic normal distribution $N\left(\mathrm{E}\left[\hat{\Theta}\right],\mathrm{V}\left[\hat{\Theta}\right]\right)$

### 3.5.8 Conclusion

In this paper, we review the newly proposed axiomatic uncertain measure theory and further introduce a measure theoretic treatment of uncertainty decision theory. We further explore the characteristics of the uncertainty decision theory. In terms of our investigations, we emphasize the fundamental mechanism of uncertainty distributions and the impacts on the general uncertainty decision processes. In the discrete uncertainty decision example, the characteristic of uncertainty distribution is intrinsic and unique because the probabilistic discrete distributions never have such features. We develop the MUP Bayes formula in continuous case.

Our efforts in this paper reveal that under general uncertainty, the decision may adopt a framework similar to statistical decision theoretic framework. However, the sub-σ-additive characteristic imposes intrinsic and unique features to the uncertainty distribution and its expectation, and thus the general uncertainty decision making is more computational demanding.

# 3.6 Bayesian Uncertainty Decision Analysis

Yanhong Cui, Renkuan Guo, Dunn Tim

Department of Statistical Sciences, University of Cape Town

Private Bag, Rhodes' Gift, Rondebosch 7701, Cape Town, South Africa

hadisna@hotmail.com renkuan.guo@uct.ac.za tim.dunne@uct.ac.za

Danni Guo

South African National Biodiversity Institute,

Private Bag X7, Claremont 7735, Cape Town, South Africa

d.guo@sanbi.org.za



This paper has been published in Reliability & Risk Analysis: Theory & Applications. (Electronic Journal of International Group on Reliability – Gnedenko E-Forum), Vol. 2, No. 1 (20), pp 70-81, March 2011. (ISSN: 1932-2321) Website: http://www.gnedenko-forum.org/Journal/

### 3.6.1 Abstract

Bayesian statistical decision theory would be questionable when applied directly to non-random uncertainty circumstances. In this paper, we investigate the basic elements of decision analysis oriented to observational data arising from a general uncertainty environment, so that a framework for Bayesian uncertainty decision doctrine is established. Further, we propose a copula-linked uncertainty marginal mechanism for constructing the uncertainty multivariate distributions to represent both observational data and an uncertainty parameter vector. This mechanism paves the way towards the establishment of an uncertainty posterior distribution of the parameter vector given the observational data, based on uncertain measure Axiom 5. Finally, we present an illustrative example of the development of a posterior uncertainty distribution for a parameter given a single observation, step by step. The significance of this paper is to establish for the first time a Bayesian uncertainty data inference and decision framework, which constitutes a critical step towards the establishment of uncertainty statistics and a Bayesian uncertainty decision theory.

### 3.6.2 Introduction

Any applied mathematical model is proposed to reflect a particular aspect of the real natural world. Decision making moving from analysis of the collected data (information) to reach a final decision is actually a process to resolve the uncertainty being faced. In the real world, there are many forms of uncertainty surrounding us, but thus far we may only deal successfully with uncertainty as randomness or fuzziness within information. How should we solve the problems with other kind of uncertainty in real business life? For example, recently, a “Made in Japan” crisis was triggered by a Toyota Prius brake fault event and quickly spread widely over other industries widely. At the first glance, it may seem a trivial event has been exaggerated by journalists. It is a well-known fact that Japanese manufacturing arms itself to the teeth with statistical quality control. There is no reason to ascribe the fault event to an absence of total quality management. Nevertheless, we cannot deny what happened, and infer that the event indicates that some unaddressed problem exists there. The only possible answer is the methodology used to manage the quality imperative does not match the real quality problem faced. In other words, while the existing quality control and decision making doctrine, which is based on probability theoretical foundation, addressing random uncertainty problems, is powerful, nevertheless for other forms of uncertainty problems, the existing theory and methodologies may be inadequate. The law of the real world tells us that each specific form of uncertainty must be addressed by the corresponding specific uncertainty doctrine and methodology. There is no universal law for addressing all the forms of uncertainties.

In this paper, we first review the basic elements of Liu's uncertain measure theory (Liu, 2007, 2009, 2010) in Section 3.6.3, and further investigate a copula-linked uncertainty marginal approach, to construct multivariate uncertainty distributions. The purpose is to represent both observational data and an uncertainty parameter vector. In Section 3.6.4, we note the basic elements of an observational-data oriented decision analysis under general uncertainty environments, in contrast to probabilistic Bayesian decision theory (Lee, 1989; Cheng, 1981; Bernardo and Smith, 1994), in order to establish a framework for a Bayesian uncertainty decision theoretic foundation. In Section 3.6.5, we propose a method to construct a posterior uncertainty distribution of parameter vector given the observational data, in terms of uncertain measure Axiom 5, see Liu (2010). In Section 3.6.6 we present an illustrative example, namely the development of a posterior uncertainty distribution for a parameter given a single observation, step by step. Section 3.6.7 concludes this paper.

## 3.6.3 Uncertain measure foundation

Uncertain measure (Liu, 2007, 2009, 2010) is an axiomatically defined set function mapping from a $\sigma$-algebra of a given space (set) to the unit interval [0, 1], which provides a measuring grade system of an uncertain event (a reflection of an uncertainty phenomenon) and enables the formal definition of an uncertain variable and its uncertainty distribution.

Let $\Xi$ be a nonempty set (space), and $\mathfrak{A}(\Xi)$ the $\sigma$-algebra on $\Xi$. Each element, let us say, $A \subset \Xi$, $A \in \mathfrak{A}(\Xi)$ is called an uncertain event. A number denoted as $\lambdabar\{A\}$, $0 \le \lambdabar\{A\} \le 1$, is assigned to the event $A \in \mathfrak{A}(\Xi)$, which indicates the uncertain measuring grade with which event $A \in \mathfrak{A}(\Xi)$ occurs. The normed set function $\lambdabar\{A\}$ satisfies following axioms given by Liu (2007, 2009, 2010):

**Axiom 1:** (Normality) $\lambdabar\{\Xi\} = 1$.

**Axiom 2:** (Monotonicity) $\lambdabar\{\cdot\}$ is non-decreasing, i.e., whenever $A \subset B$, $\lambdabar\{A\} \le \lambdabar\{B\}$.

**Axiom 3:** (Self-Duality) $\lambdabar\{\cdot\}$ is self-dual, i.e., for any $A \in \mathfrak{A}(\Xi)$, $\lambdabar\{A\} + \lambdabar\{A^c\} = 1$.

**Axiom 4:** ($\sigma$- Subadditivity) $\lambdabar\left\{\bigcup_{i=1}^{\infty} A_i\right\} \le \sum_{i=1}^{\infty} \lambdabar\{A_i\}$ for any countable event sequence $\{A_i\}$.

**Axiom 5:** (Product Measure) Let $\Xi_k, \mathfrak{A}_{\Xi_k}, \lambda_k$ be the $k^{th}$ uncertain space, $k = 1, 2, \cdots, n$. Then product uncertain measure $\lambda$ on the product measurable space $\Xi, \mathfrak{A}_{\Xi}$ is defined by

$$\lambda = \lambda_1 \wedge \lambda_2 \wedge \cdots \wedge \lambda_n = \min_{1 \le k \le n} \{\lambda_k\} \tag{1}$$

where

$$\Xi = \Xi_1 \times \Xi_2 \times \cdots \times \Xi_n = \prod_{k=1}^{n} \Xi_k \tag{2}$$

and

$$\mathfrak{A}_{\Xi} = \mathfrak{A}_{\Xi_1} \times \mathfrak{A}_{\Xi_2} \times \cdots \times \mathfrak{A}_{\Xi_n} = \prod_{k=1}^{n} \mathfrak{A}_{\Xi_k} \tag{3}$$

That is, for each product uncertain event $\Lambda \in \mathfrak{A}_{\Xi}$ (i.e, $\Lambda = \Lambda_1 \times \Lambda_2 \times \cdots \times \Lambda_n \in \mathfrak{A}_{\Xi_1} \times \mathfrak{A}_{\Xi_2} \times \cdots \times \mathfrak{A}_{\Xi_n} = \mathfrak{A}_{\Xi}$), the uncertain measure of the event $\Lambda$ is

$$\lambda\{\Lambda\} = \begin{cases} \sup\limits_{A_1 \times \cdots \times A_n \subset \Lambda} \min\limits_{1 \le k \le n} \lambda\{\Lambda_k\} & \text{if } \sup\limits_{A_1 \times \cdots \times A_n \subset \Lambda} \min\limits_{1 \le k \le n} \lambda\{\Lambda_k\} > 0.5 \\ 1 - \sup\limits_{A_1 \times \cdots \times A_n \subset \Lambda^c} \min\limits_{1 \le k \le n} \lambda\{\Lambda_k\} & \text{if } \sup\limits_{A_1 \times \cdots \times A_n \subset \Lambda^c} \min\limits_{1 \le k \le n} \lambda\{\Lambda_k\} > 0.5 \\ 0.5 & \text{otherwise} \end{cases} \tag{4}$$

**Definition 1:** (Liu, 2007, 2009, 2010) Any set function $\lambda : \mathfrak{A} \; \Xi \; \to \; 0,1$ which satisfies Axioms 1-4 is called an uncertain measure. The triple $\Xi, \mathfrak{A} \; \Xi \; , \lambda$ is called the uncertain measure space.

**Definition 2:** An uncertain variable $\xi$ is a measurable mapping, i.e., $\xi : (\Xi, \mathfrak{A}(\Xi)) \to (\mathbb{R}, \mathfrak{B}(\mathbb{R}))$, where $\mathfrak{B}(\mathbb{R})$ denotes the Borel $\sigma$-algebra on $\mathbb{R} = (-\infty, +\infty)$.

**Remark 3:** The fundamental difference between a random variable and an uncertain variable is the $\sigma$-additivity: the probability measure obeys $\sigma$-additivity and the uncertain measure obeys $\sigma$-subadditivity. The way of specifying measure inevitably has impacts on the behavior of the measurable function over the triple, and hence on the mathematical characterization of the theories. For example, in contrast to probability theory, no "uncertainty density function" can be defined and then be entered into an integral of density to characterize an uncertainty distribution. Because an uncertain measure is permitted to be $\sigma$-subadditive, any set of uncertainty distributions derived from integration, being necessarily $\sigma$-additive, will necessarily be incomplete.

**Definition 4:** (Liu, 2007, 2009, 2010) The uncertain distribution $\Psi:\mathbb{R}\to[0,1]$ of an uncertain variable $\xi$ on $\Xi,\mathfrak{A}\;\Xi\;,\lambda$ is

$$\Psi_{\xi}(x)=\lambda\left\{\tau\in\Xi\,\middle|\,\xi(\tau)\le x\right\} \tag{5}$$

**Theorem 5:** (Peng and Iwamura, 2009) The necessary and sufficient conditions for a function $\Psi:\mathbb{R}\to[0,1]$ be an uncertainty distribution function is that $\Psi$ is non-decreasing function and

$$0\le\Psi(x)\le 1,\ \forall x\in\mathbb{R} \tag{6}$$

The function $\Psi$ is referred to an uncertainty distribution function.

**Remark 6:** A probability distribution $F_X(x)$ requires right-continuity and $F_X(-\infty)=0, F_X(+\infty)=1$ in addition to those requirements of the uncertainty distribution function, while an uncertainty distribution is not limited by any continuity and $\Psi_{\xi}(-\infty)=0, \Psi_{\xi}(+\infty)=1$ requirements. This relaxation enables an uncertainty distribution to model even the most complicated pattern in real world data. The following definition reveals an essential characteristic of the uncertainty distribution.

**Definition 7:** Let $\xi$ be an uncertainty variable, which takes values from a subset, denoted as $\mathbb{E}$, of the real line $\mathbb{R}$, with $n$ discontinuity points collected in an ascending order as set $\mathbb{D}=\{c_1,\cdots,c_n\}$. The uncertainty distribution, $\Psi$, of the variable $\xi$ is specified as follows:

(1) On the set $\mathbb{D}=\{c_0,c_1,\cdots,c_n\}$,

$$\begin{aligned}&\Psi(c_i-)=\psi_{i-},\ \Psi(c_i)=\psi_i,\ \Psi(c_i+)=\psi_{i+}\\ &i=1,2,\cdots,n\end{aligned} \tag{7}$$

where $\psi_{i-}<\psi_i<\psi_{i+}$, $\psi_{1-}\ge 0$, $\psi_{n+}\le 1$, $i=1,2,\cdots,n$;

(2) At the inner points of the sub-intervals $(c_{i-1},c_i)$, $i=1,2,\cdots,n$, the uncertainty distribution $\Psi$ is continuous

$$\Psi(z)=\begin{cases}\psi_{i-1+} & \text{if } z\downarrow c_{i-1}\\ \Lambda_i(z) & \text{if } z\in(c_{i-1},c_i)\\ \psi_{i-} & \text{if } z\uparrow c_i\end{cases} \tag{8}$$

where the function $\Lambda_i$ is positive, non-decreasing, and bounded by $\psi_{i-1+}$ and $\psi_{i-}$, i.e., $\psi_{i-1+}\le\Lambda_i\le\psi_{i-}$, $i=1,2,\cdots,n$. Then $\Psi$ is an uncertainty distribution of the essential form and $\xi$ is called an essential uncertain variable.

**Remark 8:** The aim of this paper is to develop an observational-data oriented decision making doctrine. Whenever an observation is obtained, this specific observation should not be regarded as an isolated real number (or a real-valued vector), rather, it should be regarded as a representative from a population typically specified by a hypothesized uncertainty distribution. This approach matches the standard viewpoint in the statistical community. It is also a convention that the term “population” is equivalent to the term distribution, or to the term random variable. In the new uncertainty theory, this statistical convention continues. We formally state this convention as a definition on observational data.

**Definition 9:** An observation is a real number, (or more broadly, a symbol, or an interval, or a real-valued vector, a statement, etc), which is a representative of a population or equivalently of an uncertainty distribution under a given scheme comprising set and $\sigma$-algebra.

**Remark 10:** The uncertainty distribution is unknown but exists objectively. A workable solution is to hypothesize a family of uncertainty distributions of a specified functional form with unknown parameter $\theta$, where the family is denoted by $\{\Psi_{\xi}^{\theta}, \theta \in \Theta\}$.

**Definition 11:** (Liu, 2007, 2009, 2010) Let multivariate uncertainty variable $(\xi_1, \xi_2, \cdots, \xi_d)$ be defined on an uncertain measure space $(\Xi, \mathfrak{A}(\Xi), \lambda)$, then the multivariate function $\Psi_{\xi_1,\xi_2,\cdots,\xi_d} : D \to [0,1]$ is called an multivariate uncertainty distribution if

$$\Psi_{\xi_1,\xi_2,\cdots,\xi_d}(x_1, x_2, \cdots, x_d) = \lambda(\xi_1 \le x_1, \xi_2 \le x_2, \cdots, \xi_d \le x_d) \tag{9}$$

To present a concrete form of a multivariate uncertainty distribution, Guo et al [4] propose a copula-linked uncertainty marginal approach.

**Definition 12:** Let $(\xi_1, \xi_2, \cdots, \xi_d)$ be a multivariate uncertainty variable with joint uncertainty distribution $\Psi_{\xi_1,\xi_2,\cdots,\xi_d}(x_1, x_2, \cdots, x_d)$, in which all the marginal uncertainty distributions $\Psi_{\xi_1}(\cdot), \Psi_{\xi_2}(\cdot), \cdots, \Psi_{\xi_d}(\cdot)$ exist and are regular (i.e., $\Psi_{\xi_i}^{-1}(\cdot)$ exists, $i = 1, 2, \cdots, d$). Then the uncertainty copula is defined by

$$C(\Psi_{\xi_1}(x_1), \Psi_{\xi_2}(x_2), \cdots, \Psi_{\xi_d}(x_d)) = \Psi_{\xi_1,\xi_2,\cdots,\xi_d}(x_1, x_2, \cdots, x_d) \tag{10}$$

We use a bivariate uncertainty distribution as an illustrative multivariate example.

**Example 13:** Let bivariate uncertainty variable $(\xi_1, \xi_2)$ have marginal uncertainty distributions $\Psi_{\xi_1}(\cdot)$ and $\Psi_{\xi_2}(\cdot)$ respectively. The Farlie-Gumbel-Morgenstern (FGM) copula is defined by

$$C(u_1,u_2) = u_1u_2\left(1+\varpi(1-u_1)(1-u_2)\right),\ \varpi\in[-1,1] \tag{11}$$

Further, let the bivariate uncertainty variable $(\xi_1,\xi_2)$ have marginal uncertainty distributions $\Psi_{\xi_1}(\cdot)$ and $\Psi_{\xi_2}(\cdot)$ respectively, where

$$\Psi_{\xi_i}(x_i) = \frac{1}{1+\exp\left(-\frac{\pi}{\sqrt{3}\sigma_i}(x_i-\theta_i)\right)},\ i=1,2 \tag{12}$$

Then the bivariate FGM-Normal joint uncertainty distribution is

$$\Psi_{\xi_1,\xi_2}(x_1,x_2) = \prod_{i=1}^{2}\frac{1}{1+\exp\left(-\frac{\pi}{\sqrt{3}\sigma_i}(x_i-\theta_i)\right)}\left(1+\varpi\prod_{i=1}^{2}\frac{\exp\left(-\frac{\pi}{\sqrt{3}\sigma_i}(x_i-\theta_i)\right)}{1+\exp\left(-\frac{\pi}{\sqrt{3}\sigma_i}(x_i-\theta_i)\right)}\right) \tag{13}$$

Finally, it is necessary to prepare the uncertainty expectation and the variance of an uncertainty variable to support the development of an uncertainty decision doctrine.

**Definition 14:** (Liu, 2007, 2009, 2010) Let $\xi$ be an uncertainty variable defined on the uncertain space $(\Xi,\mathfrak{A}(\Xi),\lambda)$, then the expectation of $\xi$ is

$$\mathrm{E}[\xi] = \int_0^{+\infty}\lambda\{\xi\geq r\}dr - \int_{-\infty}^{0}\lambda\{\xi\leq r\}dr \tag{14}$$

provided at least one of the two integrals is finite.

**Definition 15:** (Liu, 2007, 2009, 2010) Let $\xi$ be an uncertainty variable with finite expectation $\mathrm{E}[\xi]$, then the variance of $\xi$ is

$$\mathrm{V}[\xi] = \mathrm{E}\left[(\xi-\mathrm{E}[\xi])^2\right] \tag{15}$$

**Theorem 16:** Let $\xi$ be an uncertainty variable on uncertain measure space $(\Xi,\mathfrak{A}(\Xi),\lambda)$ and $h$ be a monotonic non-decreasing function $h:\mathbb{R}\to\mathbb{R}^+$, then the expectation of $h(\xi)$ is

$$\mathrm{E}[h(\xi)] = \int_0^{+\infty}h(r)\lambda\{\xi\geq r\}dr - \int_{-\infty}^{0}h(r)\lambda\{\xi\leq r\}dr \tag{16}$$

### 3.6.4 Elements of Bayesian decision theory

A decision theory is built upon a mathematical foundation, which provides a framework (or guidelines) for decision making according to a specified criterion, based on the observational data with a distribution of the assumed uncertainty type, e.g.,

1. The statistical decision is based on probability (measure) theory, which addresses the random uncertainty;
2. The fuzzy decision theory deals with fuzziness;
3. The uncertainty decision theory deals with a general uncertainty different from randomness or fuzziness.

Recall that the statistical decision theory is established on the axiomatic foundation of probability measure.

The basic elements of statistical decision are: (1) Sample space and distributional family; (2) Decision space; (3) Loss function and decision function.

It is necessary to point out the basic elements, namely *state, action,* and *loss* in statistical decision theory (Lee, 1989; Cheng, 1981; Bernardo & Smith, 1994), are still the essential elements in the Bayesian uncertainty decision theory.

Firstly, in statistical decision theory, the state, termed "state of nature" is regarded as objectively in existence, at least in some consensus sense. In contrast, in any general uncertainty environments, the state may include subjective, judgmental or even phenomenological events or factors. Note here the conceptual interpretations that *state* acquires across the decision environments, i.e., "reality" in front of the decision makers, along with possible virtual actions, and virtual loss. The differentiation between the *state of nature* in the statistical decision theory and the *state* in the uncertainty decision theory is critical. The former reflects more or less reflecting the "truth" for the frequentist school, while the uncertainty decision theory is a mixture of subjective and objective reflections.

Secondly, the connotations of *action* in the uncertainty decision theory is virtual, in that some elements are of a precautionary nature and do not correspond to any specific state element. The nature of the mapping is from multiple states to multiple actions.. However, the inclusion of virtual action elements is extremely important, because the top decision maker does not need to deal with routine decisions of day-to-day operations but with the extreme event(s) or the most important event decision(s).

Thirdly, the *loss* mechanism in both decision theories is the same. An uncertainty decision is a selection, which minimizes the loss function $l(\theta,a)$ of an action $a$ from action space $\mathbb{A}$ for given state $\theta$ in the state space $\Theta$. However, the social loss and environmental loss extract more and more attention from the public, NGO's and the governmental agencies. In the new uncertainty decision theory, the safety factor state, the health factor state, and the environmental factor state should be automatically assigned uncertain measure grades because of their intrinsic features. In the uncertainty decision theory, an action is made in terms of observational data, denoted by $x$, which is described by an uncertainty distribution $\Psi(x\,|\,\theta)$. Based on observational data $x$ (i.e., representative of population $\Psi(x\,|\,\theta)$), a decision is actually a mapping from data space $\mathbb{D}$ into action space $\mathbb{A}$. In other words,

$$a:\mathbb{D}\rightarrow\mathbb{A} \tag{17}$$

which can be expressed by

$$a=d(x) \tag{18}$$

The loss $l(\theta,d(x))$ is measurable on the joint uncertainty space.

**Definition 17:** The expected value of the loss with respect to the uncertainty distribution of observational data $x$

$$R(\theta,d)=\mathrm{E}_{\theta}\left[l(\theta,d(x))\right] \tag{19}$$

is called a risk function.

The uncertainty distribution of observational data $x$ depends on state $\theta$, because the dependence of $R(\theta,d)$ on $\theta$ enters explicitly from $l(\theta,a)$ and also through the state $\theta$ in the distribution function $\Psi(x\,|\,\theta)$ for $x$. Therefore the uncertainty distribution of the data determines the fundamental characteristics of observational-data oriented uncertainty decision theory, which deserves further exposure.

Let us consider the uncertainty decision problem for a given uncertainty distribution. Assume a state space $\Theta=\mathbb{R}$, and a continuous action space $\mathbb{A}=\mathbb{R}$, and the loss function defined by

$$l(\theta,a)=w(\theta)(\theta-a)^2 \tag{20}$$

i.e., a quadratic loss function is assumed

**Definition 18:** (Uncertainty Bayes loss) Given a continuous state space $\Theta$, the uncertainty variable $\theta$ is defined on uncertain space $\left(\Theta,\mathfrak{B}(\Theta),\lambda_{\theta}\right)$, where $\lambda_{\theta}\left(\cdot\right)$ is an uncertain measure. The uncertain distribution $\Psi_{\theta}\left(\cdot\right)$ is defined on $\left(\Theta,\mathfrak{B}(\Theta)\right)$). Then the average of loss with respect to state space for a given action $a\in\mathbb{A}$, is the quantity

$$B(a)=\mathrm{E}\left[l(\theta,a)\right]=\int_{\Theta}l(y,a)\,d\Psi_{\theta}(y) \tag{21}$$

and is called the uncertainty Bayes loss for a given action $a$.

**Definition 19:** (Uncertainty Bayes risk) The uncertainty Bayes risk is defined by

$$B(d)=\mathrm{E}\left[l\left(\theta,d(x)\right)\right]=\int_{\Theta}l\left(\theta,d(x)\right)d\Psi(\theta) \tag{22}$$

**Definition 20:** (Uncertainty Bayes rule) A Bayes decision rule, denoted as $d^{B}$, is a rule such that the Bayes risk is minimized, i.e.,

$$B\left(d^{B}\right)=\min_{d\in\mathbb{D}}\left\{B(d)\right\} \tag{23}$$

**Example 21:** Given a continuous state space $\Theta=\mathbb{R}$, the uncertainty variable $\theta$ is defined on an uncertain space $\left(\mathbb{R},\mathfrak{B}(\mathbb{R}),\lambda_{\partial}\right)$, where $\lambda_{\theta}\left(\cdot\right)$ is properly defined. The uncertainty distribution is assumed to be

$$G_{\theta}(y)=\frac{y-a}{2(b-a)}\vartheta_{[a,b]}(y)+\frac{y+c-2b}{2(c-b)}\vartheta_{[b,c]}(y) \tag{24}$$

Then we derive the average loss with respect to the state space for a given action $a\in\mathbb{A}$, as the uncertainty Bayes loss:

$$B(a)=\mathrm{E}\left[l(\theta,a)\right]=\int_{\Theta}w(y)(y-a)^{2}\,d\Psi_{\theta}(y) \tag{25}$$

Set $w(\theta)=w_{0}$, a constant, then the uncertainty Bayes loss is

$$\begin{aligned}
B(a) &= \frac{w_0}{2(\beta-\alpha)}\int_{\alpha}^{\beta}(y-a)^2 dy + \frac{w_0}{2(\gamma-\beta)}\int_{\beta}^{\gamma}(y-a)^2 dy \\
&= \frac{w_0}{2(\beta-\alpha)}\left[(\beta-a)^3-(\alpha-a)^3\right] + \frac{w_0}{2(\gamma-\beta)}\left[(\gamma-a)^3-(\beta-a)^3\right] \\
&= \frac{w_0}{2}\left[(\beta-a)^2+(\alpha-a)^2+(\beta-a)(\alpha-a)\right] + \frac{w_0}{2}\left[(\gamma-a)^2+(\beta-a)^2+(\gamma-a)(\beta-a)\right] \\
&= \frac{w_0}{2}\left(3a^2-3(\alpha+\beta)a+\alpha^2+\beta^2\right) + \frac{w_0}{2}\left(3a^2-3(\gamma+\beta)a+\gamma^2+\beta^2\right) \\
&= w_0\left(3a^2-\frac{3}{2}(\alpha+2\beta+\gamma)a+\frac{1}{2}\left(\alpha^2+2\beta^2+\gamma^2\right)\right)
\end{aligned} \tag{26}$$

With an appropriate specification of decision function in term of data, the uncertain Bayesian decision analysis can be formulated.

### 3.6.5 A posterior uncertainty distribution

When $\Theta, \mathfrak{B}_{\Theta}, \lambda_{\theta}$ is an uncertain (prior) space and $\mathfrak{X}, \mathfrak{B}_{\mathfrak{X}}, P_{\theta}$ is a probability space, we actually use random sample information to make inferences on the uncertain parameter $\theta$. The critical step in the probabilistic Bayesian inference is to develop the posterior distribution for parameter $\theta$. We strongly believe that the Bayesian uncertainty inference requires parallel manipulations.

Let $\Theta, \mathfrak{B}_{\Theta}$ be a parameter measurable space, $\mathfrak{X}, \mathfrak{B}_{\mathfrak{X}}$ be a sample measurable space.

**Definition 22:** An uncertain measure defined on $\Theta, \mathfrak{B}_{\Theta}$ is called an uncertain prior measure, denoted as $\lambda_{\theta}$. The space $\Theta, \mathfrak{B}_{\Theta}, \lambda_{\theta}$ is called an uncertain prior space, the uncertain distribution $G\ y\ = \lambda_{\theta}\ \theta \le y$ is called an uncertain prior distribution.

An uncertainty variable, denoted by $\xi$, is defined on a measurable space $\mathfrak{X}, \mathfrak{B}_{\mathfrak{X}}$ with uncertainty distributional family $\Psi_{\theta}, \theta \in \Theta$ where $\Theta$ is a parameter space. Formally,

**Definition 23:** The uncertainty observations are representatives of an uncertainty variable $\xi$, which is called an uncertainty population, or alternatively, called as an uncertainty distribution $\Psi_{\xi}(x|\theta), \theta \in \Theta$. The uncertainty variable $\xi$ is defined on $(\mathfrak{X}, \mathfrak{A}_{\mathfrak{X}}, \tilde{\lambda}_{X})$. The uncertainty distribution is

$$\Psi_{\xi}(x|\theta) = \tilde{\lambda}_{\xi}\{\xi \le x|\theta\} \tag{27}$$

**Remark 24:** The uncertainty observations are presented by observers or experts, while the prior distribution (prior uncertain measure) is offered by knowledgeable experts on the observers' behaviours. In probabilistic Bayesian statistics it is typically assumed that the prior and the likelihood are independent of each other. In Bayesian uncertainty doctrine we continue to follow this convention without any theoretical justification, although the independence between prior and likelihood is debatable.

**Remark 25:** The joint cumulative distribution of the observational data $(x_1, x_2, \cdots, x_n)$ may be specified by a hypothesized copula functional according to the features of the data, as

$$\begin{aligned}
&\Psi_{\xi_1,\xi_2,\cdots,\xi_n}\left(x_1, x_2, \cdots, x_n \middle| \theta\right) \\
&= \bar{\lambda}_{\xi_1,\xi_2,\cdots,\xi_n}\left\{\xi_1 \le x_1, \xi_2 \le x_2, \cdots, \xi_n \le x_n \mid \theta\right\} \\
&= C_{\underline{\varpi}}\left(\Psi_{\xi_1}(x_1), \Psi_{\xi_2}(x_2), \cdots, \Psi_{\xi_n}(x_n)\right)
\end{aligned} \tag{28}$$

where $\Psi_{\xi_1}(\cdot), \Psi_{\xi_2}(\cdot), \cdots, \Psi_{\xi_n}(\cdot)$ are given marginal uncertainty distributions and $\underline{\varpi}$ is unknown parameter vector. In contrast, within probability theory, the multivariate joint distribution function is $F_{X_1,X_2,\cdots,X_n}\left(x_1, x_2, \cdots, x_n \middle| \theta\right)$ . Given a population $F\left(x \middle| \theta\right)$, the *i.i.d.* random sampling observations have a joint distribution function

$$F_{X_1,X_2,\cdots,X_n}\left(x_1, x_2, \cdots, x_n \middle| \theta\right) = \prod_{k=1}^{n} F_{X_k}\left(x_k \middle| \theta\right) \tag{29}$$

Also, the joint density (i.e., the likelihood function) is

$$f_{X_1,X_2,\cdots,X_n}\left(x_1, x_2, \cdots, x_n \middle| \theta\right) = \frac{\partial}{\partial x_1}\frac{\partial}{\partial x_2}\cdots\frac{\partial}{\partial x_n} F_{X_1,X_2,\cdots,X_n}\left(x_1, x_2, \cdots, x_n \middle| \theta\right) = \prod_{k=1}^{n} f_{X_k}\left(x_k \middle| \theta\right) \tag{30}$$

Now, let us continue our arguments on the posterior uncertainty distribution of $\theta$ . For the convenience, let us assume that a pair of observations $x_1, x_2$ is obtained from the bivariate uncertainty variable, denoted by $\xi_1, \xi_2$ , which is defined by a hypothesized bivariate FGM-normal uncertainty distribution

$$\Psi_{\xi_1,\xi_2}\left(x_1, x_2\right) = \prod_{i=1}^{2} \frac{1}{1+\exp\left(-\frac{\pi}{\sqrt{3}\sigma_i}\left(x_i-\theta_i\right)\right)} \left(1+\varpi \prod_{i=1}^{2} \frac{\exp\left(-\frac{\pi}{\sqrt{3}\sigma_i}\left(x_i-\theta_i\right)\right)}{1+\exp\left(-\frac{\pi}{\sqrt{3}\sigma_i}\left(x_i-\theta_i\right)\right)}\right) \tag{31}$$

with marginals

$$\Psi_{\xi_i}\left(x_i\right) = \frac{1}{1+\exp\left(-\frac{\pi}{\sqrt{3}\sigma_i}\left(x_i-\theta_i\right)\right)},\ i=1,2 \tag{32}$$

Then the bivariate uncertainty distribution has parameter vector $\theta = \left(\theta_1, \sigma_1, \theta_2, \sigma_2, \varpi\right)$. For simplification only, we set $\theta = \theta_1 = \theta_2$, and assume that both $\sigma_0 = \sigma_1 = \sigma_2$, and $\varpi$ are known. Then what we aim to derive the posterior distribution of parameter $\theta$, i.e., $\Psi_\theta\left(y \mid x_1, x_2, \sigma_0, \varpi_0\right)$.

For the parameters, there is again a specification issue of the joint multivariate uncertainty prior distribution. For example, the FGM-normal bivariate uncertainty distribution $\Psi_{\xi_1,\xi_2}\left(x_1, x_2\right)$ has five parameters, i.e., $\theta = \left(\theta_1, \sigma_1, \theta_2, \sigma_2, \varpi\right)$. The full specification of prior vector needs a five-dimensional copula, $C_\varphi\left(v_1, v_2, v_3, v_4, v_5\right)$ with marginals $p_i\left(v_i\right), i = 1, 2, \cdots, 5$, and the prior takes a form

$$\Psi_{v_1, v_2, \cdots, v_5}\left(y_1, y_2, \cdots, y_5\right) = C_\varphi\left(p_1\left(v_1\right), p_2\left(v_2\right), \cdots, p_5\left(v_5\right)\right) \tag{33}$$

**Definition 26:** Let $\Psi\left(x_1, x_2, \cdots, x_n, \underline{\theta}\right)$ be the joint uncertainty distribution of the uncertainty observations $\left(x_1, x_2, \cdots, x_n\right)$ together with the parameter vector $\underline{\theta}$. Note the event

$$\Lambda = \left\{\underline{X} \le \underline{x}, \underline{\theta} \le \underline{y}\right\} \tag{34}$$

Then the joint distribution of $\underline{X}$ and $\underline{\theta}$ defined on $\left(\mathfrak{X}, \mathfrak{A}_{\mathfrak{X}}, \lambda_X\right)$ and $\left(\Theta, \mathfrak{A}_\Theta, \lambda_\theta\right)$ respectively, according to Axiom 5 (Liu, 2010) is the joint uncertainty measure defined by

$$\begin{aligned}
&\Psi_{\underline{X},\underline{\theta}}\left(\underline{x}, \underline{y}\right) = \lambda_{\underline{X},\underline{\theta}}\left\{\underline{X} \le \underline{x}, \underline{\theta} \le \underline{y}\right\} \\
&= \begin{cases}
\sup\limits_{A_1 \times A_2 \subset \Lambda} \min\left(\lambda_{\underline{\theta}}\left\{A_1\right\}, \lambda_{\underline{X}}\left\{A_1\right\}\right) & \text{if} \quad \text{if} \sup\limits_{A_1 \times A_2 \subset \Lambda} \min\left(\lambda_{\underline{\theta}}\left\{A_1\right\}, \lambda_{\underline{X}}\left\{A_1\right\}\right) > 0.5 \\
1 - \sup\limits_{A_1 \times A_2 \subset \Lambda^C} \min\left(\lambda_{\underline{\theta}}\left\{A_1\right\}, \lambda_{\underline{X}}\left\{A_1\right\}\right) & \text{if} \quad \text{if} \sup\limits_{A_1 \times A_2 \subset \Lambda^C} \min\left(\lambda_{\underline{\theta}}\left\{A_1\right\}, \lambda_{\underline{X}}\left\{A_1\right\}\right) > 0.5 \\
0.5 & \text{otherwise}
\end{cases}
\end{aligned} \tag{35}$$

**Definition 27:** We denote $\Psi_{\underline{X}}\left(\underline{x}\right) = \Psi_{X_1, X_2, \cdots, X_n}\left(x_1, x_2, \cdots, x_n\right)$ as the absolute joint uncertainty distribution.

$$\Psi_{X_1, X_2, \cdots, X_n}\left(x_1, x_2, \cdots, x_n\right) = \sup_{\underline{y} \in \Theta}\left(\Psi_{X_1, X_2, \cdots, X_n}\left(x_1, x_2, \cdots, x_n, \underline{y}\right|\right)\right) \tag{36}$$

For example, if a pair of bivariate FGM-normal uncertainty observation $\left(x_1, x_2\right)$ is obtained, then

$$\Psi_{X_1, X_2}\left(x_1, x_2\right) = u_1 u_2\left(1 + \varpi\left(1 - u_1\right)\left(1 - u_2\right)\right) \tag{37}$$

where

$$u_i = \Psi_{\xi_i} \; x_i \; = \frac{1}{1+\exp\left(-\frac{\pi}{\sqrt{3}\sigma_i} \; x_i - \theta_i\right)}, \; i = 1,2 \tag{38}$$

Finally, we define a Bayesian uncertainty posterior for $\underline{\theta}$.

**Definition 28:** We denote $\Psi_{\underline{\theta}}\left(\underline{y} \middle| x_1, x_2, \cdots, x_n\right)$ as the posterior uncertainty distribution under the Maximum Uncertainty Principle, The MUP posterior uncertainty distribution is thus

$$\Psi_{\underline{\theta}}\left(\underline{y} \middle| x_1, x_2, \cdots, x_n\right) = \frac{\Psi_{\xi_1,\xi_2,\cdots,\xi_n,\underline{\theta}}\left(x_1, x_2, \cdots, x_n, \underline{y}\right)}{\Psi_{\xi_1,\xi_2,\cdots,\xi_n}\left(x_1, x_2, \cdots, x_n\right)} \tag{39}$$

### 3.6.6 A Bayesian posterior uncertainty example

In this section, we take the uncertainty zigzag distribution as the uncertainty prior, and Liu's (2007, 2009, 2010) normal distribution as uncertainty observation distribution, and in a step by step manner, illustrate the construction of a posterior uncertainty distribution.

The observational data is assumed to be a representative value of the population observed, which can be specified by a hypothesized uncertainty distribution. For our example, the hypothesized uncertainty distribution is Liu's uncertainty normal distribution (Liu, 2007, 2009, 2010):

$$\Psi\left(x \middle| \theta, \sigma_0\right) = \frac{1}{1+\exp\left(-\frac{\pi}{\sqrt{3}\sigma_0}(x-\theta)\right)} \tag{40}$$

As an illustration, it is assumed that the standard deviation is given, denoted by $\sigma_0$, the only unknown is parameter $\theta$, the mean or expectation of the uncertainty distribution. Because in Bayesian treatments the parameter in the distribution function is no longer an unknown real number, the parameter is treated as an uncertainty variable, Its distribution is supposed to be uniquely specified by a given uncertainty measure $\lambda_\theta$ defined on an uncertain measurable space $\Theta, \mathfrak{A} \; \Theta$ .

In practice, the unknown parameter is usually specified by an uncertainty distribution. Although the uncertainty distribution can induce an uncertain measure on Borel measurable space $\mathfrak{Y},\mathfrak{B}\ \mathfrak{Y}$ , nevertheless, it is unique in the sense of an equivalence class. The uncertainty prior distribution is assumed to be

$$G_\theta(y)=\frac{y-a}{2(b-a)}\vartheta_{(a,b]}(y)+\frac{y+c-2b}{2(c-b)}\vartheta_{(b,c]}(y) \tag{41}$$

where

$$\vartheta_{(a,b]}(y)=\begin{cases}1 & \text{if} \quad a<y\le b\\ 0 & \text{otherwise}\end{cases} \tag{42}$$

We further assume that a single observation $x=2.3$ is taken from hypothesized Liu's uncertainty normal distribution $\Psi_X\left(x\middle|\theta,2.00\right)=1\big/\left(1+\exp\left(-\pi(x-\theta)/2\sqrt{3}\right)\right)$ and the uncertainty prior parameter $a,b,c = 0,2,3$ , i.e., the uncertainty prior distribution is

$$G_\theta(y)=\frac{1}{4}y\vartheta_{(0,2]}(y)+\frac{1}{2}(y-1)\vartheta_{(b,c]}(y) \tag{43}$$

The Axiom 5 based posterior uncertainty distribution of $\theta$ given uncertainty observation $x=2.3$ is

$$\Psi_\theta\left(y\middle|x=2.3,\sigma_0=2\right)=\frac{\Psi_{\xi,\theta}(2.3,y)}{\sup\limits_{y\in\Theta}\left(\Psi_{\xi,\theta}(2.3,y)\right)} \tag{44}$$

where

$$\Psi_{\xi,\theta}(2.3,y)=\begin{cases}\sup\limits_{A_1\times A_2\subset\Lambda}\min\left(G_\theta(y),\Psi_\xi\left(2.3\middle|y,2.0\right)\right) & \text{if} \quad \sup\limits_{A_1\times A_2\subset\Lambda}\min\left(G_\theta(y),\Psi_\xi\left(2.3\middle|y,2.0\right)\right)>0.5\\ 1-\sup\limits_{A_1\times A_2\subset\Lambda^C}\min\left(G_\theta(y),\Psi_\xi\left(2.3\middle|y,2.0\right)\right) & \text{if} \quad \sup\limits_{A_1\times A_2\subset\Lambda^C}\min\left(G_\theta(y),\Psi_\xi\left(2.3\middle|y,\sigma_0\right)\right)>0.5\\ 0.5 & \text{otherwise}\end{cases} \tag{45}$$

and

$$\sup_{y\in\Theta}\left(\Psi_{\xi,\theta}(2.3,y)\right)=0.60392 \tag{46}$$

The plot of the posterior uncertainty distribution $\Psi_\theta\left(y\,|\,x=2.3,\sigma_0=2\right)$ is shown in Figure 3.6.1.

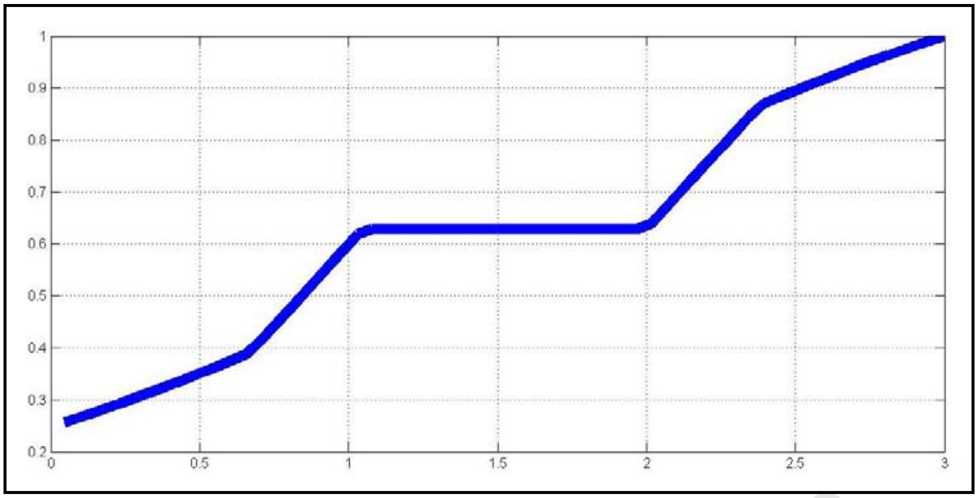



Figure 3.6.1 Posterior uncertainty distribution $\Psi_\theta\left(y \middle| x = 2.3, \sigma_0 = 2\right)$

Once we find the posterior uncertainty distribution $\Psi_\theta\left(y \middle| x = 2.3, \sigma_0 = 2\right)$, it is very natural to calculate the uncertainty posterior mean $\mathrm{E}\ \theta \mid x = 2.3$ and variance $V\ \theta \mid x = 2.3$ , and carry on further Bayesian inferential analysis.

### 3.6.7 Conclusions

In this paper, a Bayesian uncertainty decision theoretical framework is proposed under the uncertain measure foundation, which paves the way toward data-oriented inferential uncertainty statistics.

The contributions of this paper are listed as follows:

(1) for the first time a concrete uncertainty multivariate uncertainty distribution, in terms of an uncertainty copula with uncertainty marginals, is presented in the uncertainty theory literature;

(2) for the first time an uncertainty product measure data-oriented posterior uncertainty distribution is developed from axiom 5; and

(3) a detailed illustrative example is given in stepwise manner.

Definitely, the treatments in this paper are debatable. Particularly, the independence between prior measure and observational data uncertainty distribution measure may be contested. Also, we have not explored the necessary and sufficient conditions for an uncertainty copula constructed multivariate uncertainty distribution to be uncertainty measure. There are many unaddressed questions ahead of us for the new Bayesian uncertainty decision theory to become fully applicable in industries and business.

## 3.7 Probabilistic DEAR Models

Yanhong Cui, Renkuan Guo

Department of Statistical Sciences, University of Cape Town

Private Bag, Rhodes' Gift, Rondebosch 7701, Cape Town, South Africa

hadisna@hotmail.com renkuan.guo@uct.ac.za

Danni Guo

South African National Biodiversity Institute,

Private Bag X7, Claremont 7735, Cape Town, South Africa

d.guo@sanbi.org.za



This paper has been submitted to the International Journal of Machine Learning and Cybernetics, 2011.

### 3.7.1 Abstract

The DEAR (differential equation associated regression) is a flexible and powerful data mining modeling approach, which is intended to catch up the first-order non-linear trend (i.e., regularity) governing the behavior of the data under investigation. DEAR modeling is a formal mathematical-statistical representation of the so-called grey differential equation model. It should be pointed out that DEAR models were originally proposed on the random fuzzy theoretical foundation. Nevertheless, DEAR models can be defined on any measure theoretic platform, for example, probabilistic, fuzzy, or uncertain measure foundation as long as the model and approximation two constituting components are appropriately specified. In this paper, we re-examine the compositional elements of DEAR models and the potential model selection portfolio in the statistical machine learning (SML) algorithm developments. Then the differential equation backed DEAR may contribute to the statistical machine learning algorithm significantly, particularly, in developing robot movement system, where the motion laws are expressed directly by a set of differential equations. Under a statistical decision theoretical framework, a DEAR model which is constituted by a random function with a linear difference equation-wise regression as the central tendency and a variance bound specified by Gaussian error analysis theory is developed delicately, in which the prior distribution will be facilitated by a Gaussian process such that the replication of sampling for estimating the weight matrix will be avoided. We not only address the model selection compositional elements of the statistical machine learning algorithm but also address the optimization scheme, which is called λ-global optimization scheme to make the DEAR learning as one of the fastest, most efficient and accurate SML algorithm.

**Keywords:** SML Algorithm, Grey Differential Equation, Coupling Principle, DEAR models, Gaussian process DEAR models, λ-global optimization scheme

### 3.7.2 Introduction

Since machine learning, a merging field of computer science and statistics, was initiated by a group of computer scientists fifty years ago, it has become one of the fastest developing and most successful scientific fields with many magnificent industrial applications, for example, genetic engineering, speech recognition, face recognition, computer vision, Bio-surveillance, robot control, etc, and the new computational statistics, i.e., statistical machine learning (abbreviated as

SML) algorithm, has accelerating the developments of many empirical sciences (Abe, 2005; Abe and Inoue, 2001; Allen et al., 2006; Bao and Intille, 2004; Billard and Diday, 2000; Cucchiara et al. 2005).

How a computer system can automatically improve itself with experiences it learned from data and what are the fundamental laws governing the learning process are the central theme of statistical machine learning field, (Domingos and Pazzani, 1997; Hsu and Lin, 2002; Kearns and Vazirani, 1994; Koggalage and Halgamuge, 2004; Mierswa et al., 2006; Witten and Frank (2005). Statistical machine learning algorithms take a training set, form hypotheses or models, and make predictions about the future. Because the training set is finite and the future is non-determinate, a learning algorithm usually does not yield absolute guarantees of performance of the computation schemes. Instead, probabilistic bounds on the performance of machine learning schemes are quite common. Probably approximately correct learning (PAC learning), proposed by Leslie and Valiant (1984), which allows "accurate" mathematical analysis of learning.

Note that the SML algorithm is a high-level concept being composed of many lower level concepts, which might be also called xx algorithm, for example, in the computer science, it is often seen the usage of genetic algorithm (GA), the ant colony algorithm, or the monkey algorithm etc. But those optimization algorithms can only be compositional element of the SML algorithm. In order to distinguish the two level algorithms, we keep the SML algorithm usage but call those optimization algorithms as optimization schemes, for example GA scheme.

The differential equation associated regression (abbreviated as DEAR) model is an extension to classical linear regression models (Chen, 1981; Draper and Smith, 1966; Myers, 2000; Rao, 1973), by coupling a differential equation model and a (corresponding) regression model together for better catching the trends underlying the data investigated, see Guo et al. (2006). DEAR is actually a grey differential equation proposed by Deng (1984, 1985, 2002, 2004) and widely developed by Liu and Lin (2006), and Wen (2004), but the formation or the representation of DEAR is rigorous in mathematics, see Guo et al. (2009).

Although DEAR model is rigorous in its theoretical treatments, the random fuzzy regression feature prevents practical data analysts, field engineers and businessmen as well as governmental employees from applying the DEAR models (Guo et al., 2007; Guo et al., 2008; Guo and Guo, 2009). A natural question appears in front of us: is it possible that DEAR may devote itself to facilitate a new regression analysis which enjoys mathematical rigor and at the same time to make the modeling as simple as possible in probabilistic foundation. It is further to be aware that without the probabilistic treatments, DEAR modeling family will not merge into current statistical machine learning model main stream rapidly.

It is well-accepted scientific fact that the Newton Motion Laws are presented by a set of first or second differential equations, therefore differential equation backed DEAR models have a unique merit in developing motion control system and thus the statistical machine learning algorithm, which should be an integrated part of the motion control system.

We further noticed in our DEAR modeling exercises that the most popular GA searching scheme did not always offer the optimal solution and sometimes its computation speed is too slow. Therefore, developing faster optimization scheme is also an integrated part of merging DEAR models into the SML algorithm, and hence the control system. The λ-scheme is just born to serve the DEAR computations.

In this paper, we base on statistical decision theory to develop a new probabilistic DEAR model, which is effectively utilizes the members from a Gaussian process as the error terms (Rasmussen and Williams, 2006; Snelson et al., 2004; Williams and Rasmussen, 1996). Hence, the autocovariance matrix can be facilitated part of the DEAR error structure. That is, the Gaussian process DEAR model will merge data-design matrix and the intrinsic autocovariance matrix together to address the regression coefficient estimation problem.

The new DEAR regression model will be accessible to any data mining algorithms and is suitable to computer system self learning and adjusting because DEAR algorithm becomes a weighted regression, while the weight matrix is just the intrinsic autocovariance matrix plus the Gaussian approximation error matrix. The dimension of the weight matrix depends upon data size, and the coefficient estimator depends on the data-design matrix $X$ and weight matrix $\Sigma$, i.e., $X\Sigma^{-1}X$.

The structure of the remainder of this paper is as follows: Section 3.7.3 provides the review on the basic elements in SML theory; in Section 3.7.4 we review grey differential equation and its DEAR formation; Section 3.7.5 contributes to Gaussian process DEAR model developments; in Section 3.7.6, the λ-global optimization scheme is introduced so that we devote our efforts in this paper toward SML algorithm from model selection and computational scheme two aspects, and the last Section 3.7.7 offers a brief concluding remark.

### 3.7.3 The Elements in SML Algorithm

Because of the statistical machine learning (SML) was initiated and dominated by the computer science researchers in the last fifty years, the majority of its terminologies are unfamiliar to statisticians inevitable. However, an appropriate "translation" can gap the difference between the two sides and let this merging field benefit more from both sides and speed up the developments of SML algorithm.

For example, probably approximately correct (abbreviated as PAC) algorithm is the simplest one in the machine learning algorithm. PAC is the language of computer science researchers.

**Definition 1:** Concept class $\mathcal{C}$ is PAC-learnable if there exists a learning algorithm $\mathcal{L}$ such that for all $c \in \mathcal{C} \quad \varepsilon > 0, \quad \delta > 0,$ and all distributions $\mathcal{D}$

$$P\left\{\mathrm{error}_{\mathcal{D}}\left(h_{\mathcal{S}}\right) \le \varepsilon \mid \mathcal{S} \overset{d}{\sim} \mathcal{D}\right\} \ge 1-\delta \tag{1}$$

for a random sample $\mathcal{S}$ of size $m = poly\left(\varepsilon^{-1}, \delta^{-1}\right)$ which is a given polynomial.

**Remark 2:** When a concept class $\mathcal{C}$ is engaged, it is assumed that concept class $\mathcal{C}$ is known to the learning algorithm $\mathcal{L}$. The concept of "example" in SML is actually the subset of "sampling data" in statistics. Whenever saying "training and test examples are drawn from distribution $P$", it means that the random sample is drawn from population $P$ identically independently distributed (*i.i.d.*) in statistics. The term "probably' is simply meaning an estimator possesses confidence level $1-\delta$, while the term "approximately correct" indicates the approximation accuracy is not less than $1-\varepsilon$. We list the PAC learning notations and give the corresponding statistical connotations:

| | |
|---|---|
| $\mathcal{X}$ | Instance set, $\mathcal{X} = \{x \text{ is an instance or example}\}$, which is called a sample space in statistics; |
| $c$ | Target concept $c : \mathcal{X} \to \{0,1\}$, which is a decision rule or a random mapping; |
| $\mathcal{C}$ | Concept set $\mathcal{C} = \{c\}$, which is the action space or mapped space in statistical decision theory; |
| $P$ | A given probability distribution over $\mathcal{X}$, i.e., a probability space $(\mathcal{X}, \mathcal{F}_{\mathcal{X}}, P)$; |
| $\mathcal{H}$ | Concept hypothesis set $\mathcal{H} = \{h : \text{ concept hypothesis}\}$, concept hypothesis is actually a given statement or a function which is close to a target concept $c$; |
| $\mathcal{L}$ | A learning algorithm receives sample $\mathcal{S}$ and selects a hypothesis $h_{\mathcal{S}}$ from |

$\mathcal{H}$ approximating $c$. It is obvious that is a composite statistical proposition, which may be converted into a composite statistical hypothesis testing, $H_0: \theta = \theta_0$ vs $H_1: \theta \neq \theta_0$.

PAC-learning algorithm theory provides a distribution-free convergence guarantees with polynomially bounded sample sizes (Haussler, 1992). Mathematically, the PAC-Bayes formula has been developed accordingly.

**Theorem 3:** (MacAllester, 1999) Let

| | |
|---|---|
| $p(\omega)$ | Prior (independent of $\mathcal{S}$, random samples); |
| $Q(\omega \mid \mathcal{S})$ | Posterior of $\omega$ given $\mathcal{S}$; |
| $\mathrm{G}(Q(\omega \mid \mathcal{S}))$ | Expected generalization error $\mathrm{G}(Q(\omega \mid \mathcal{S})) = \mathrm{E}_{Q(\omega \mid \mathcal{S})}[g(\omega)]$; |
| $\mathrm{EEM}(\mathcal{S}, Q)$ | Expected empirical error $\mathrm{E}EMP(\mathcal{S}, Q) = \mathrm{E}_{Q(\omega \mid \mathcal{S})}[EMP(\mathcal{S}, \omega)]$; |
| $\mathrm{D}[Q \Vert P]$ | Relative entropy $\mathrm{D}[Q \Vert P] = \mathrm{E}_{Q(\omega \mid \mathcal{S})}\left[\log\left(\frac{dQ(\omega \mid \mathcal{S})}{dP(\omega)}\right)\right]$. |

Then

$$P\left\{\mathrm{D}_{Ber}\left[EMP(\mathcal{S}, Q) \Vert g(Q)\right] \geq \frac{1}{n}\left(\mathrm{D}[Q \Vert P] + \log\left(\frac{n+1}{\delta}\right)\right)\right\} \leq \delta \tag{2}$$

"These bounds, however, are notoriously loose and impractical", as Haussler pointed out, see [39], and [40].

Just as the Laffery and Wasserman (2006) pointed out, "Overall, the PAC model's focus on the traditional complexity-theoretic dividing line of polynomial versus exponential time or space has resulted in the theory being largely built up around negative examples. This suggests that the underlying theoretical framework may be too rigid. It would be very interesting to develop new theoretical frameworks based on the *tradeoff* between computation and risk that is important in practice; this tradeoff appears to have largely been ignored in both statistical theory and computational learning theory".

Therefore it is inevitable to consider other theoretical ground for facilitating a better formulation with reasonable tradeoff between computational speed and risk. Professor Haussler proposed a generalized decision theoretic framework. For a better understanding, let us review the basic elements of statistical decision theory first and then discuss Haussler's six elements.

The basic elements of statistical decision are: (1) Sample space and distributional family; (2) Decision space; (3) Loss function and decision function.

It is necessary to point out the basic elements, namely *state, action,* and *loss* in statistical decision theory (Abe, 2005; Abe and Inoue, 2001; Allen, 2006), are still the essential elements in the SML algorithm.

In the statistical decision theory, an action is made in terms of observational data, denoted by $x$, which is governed by an probability distribution function $F(x\,|\,\theta)$. Based on observational data $x$ (i.e., representative of statistical population $F(x\,|\,\theta)$), a decision is actually a mapping from data space $\mathbb{D}$ into action space $\mathbb{A}$. In other words,

$$a:\mathbb{D}\to\mathbb{A} \tag{3}$$

which can be expressed by

$$a=d\left(x\right) \tag{4}$$

The loss $l\big(\theta,d\left(x\right)\big)$ is measurable on the joint probability space $\left(\Theta\times\mathcal{X},\mathcal{F}_{\Theta\times\mathcal{X}},P\right)$. Obviously, Haussler's generalized decision theory does not escape from sample space and sample distribution issues.

**Definition 4:** The expected value of the loss with respect to the probability distribution of observational data $x$

$$R\left(\theta,d\right)=\mathrm{E}_{\theta}\left[l\left(\theta,d\left(x\right)\right)\right] \tag{5}$$

is called a risk function.

The probability distribution of observational data $x$ depends on state $\theta$, because the dependence of $R(\theta,d)$ on $\theta$ enters explicitly from $l(\theta,a)$ and also through the state $\theta$ in the distribution function $F(x\,|\,\theta)$ for $x$. Therefore the probability distribution of the data determines the fundamental characteristics of observational-data oriented statistical decision theory, which deserves further exposure.

Let us consider the statistical decision problem for a given probability distribution. Assume a state space $\Theta=\mathbb{R}$, and a continuous action space $\mathbb{A}=\mathbb{R}$, and the quadratic loss function is defined by

$$l(\theta,a)=w(\theta)(\theta-a)^2 \tag{6}$$

**Definition 5:** (Bayes loss) Given a continuous state space $\Theta$, the uncertainty variable $\theta$ is defined on uncertain space $(\Theta,\mathcal{B}_\Theta,P_\theta)$, where $P_\theta(\cdot)$ is a probability measure. The probability distribution $F_\theta(\cdot)$ is defined on $(\Theta,\mathcal{B}_\Theta)$. Then the average of loss with respect to state space for a given action $a\in\mathbb{A}$, is called the Bayes loss for a given action *a*:

$$B(a)=\mathrm{E}\left[l(\theta,a)\right]=\int_\Theta l(y,a)dF_\theta(y) \tag{7}$$

**Definition 6:** (Bayes rule) A Bayes decision rule, denoted as $d^B$, is a rule such that the Bayes risk is minimized, i.e.,

$$B\left(d^B\right)=\min_{d\in\mathbb{D}}\left\{B(d)\right\} \tag{8}$$

Now, let us further examine the six elements in Haussler's the generalized statistical decision theoretic framework for the SML algorithm.

| | |
|---|---|
| $\mathcal{X}$ | $\mathcal{X}=\{x:\text{ all possible instances or examples}\}$; |
| $\mathcal{Y}$ | The outcome space |
| $\mathcal{A}$ | The decision space |
| $\mathcal{H}$ | The decision rule space $\mathcal{H}=\{h:\mathcal{X}\to\mathcal{A}\}$ |
| $\mathcal{P}$ | $\mathcal{P}=\{P:\mathcal{X}\times\mathcal{Y}\to\mathcal{P}\}$ A family of joint distributions |
| $l$ | $l:\mathcal{Y}\times\mathcal{A}\to\mathbb{R}$, $l(y,a)$, loss function |

In statistician's language, it simply says that random examples are drawn independently from the same population (a given probability distribution $P\in\mathcal{P}$ defined on the sample space $\mathcal{Z}=\mathcal{X}\times\mathcal{Y}$), i.e., a random sample $S$ is drawn.

**Definition 7:** The Haussler's risk function is defined by

$$r_{h,l}(P)=\mathrm{E}\left[l\left(y,h(x)\right)\right]=\int_Z l\left(y,h(x)\right)dP(x,y) \tag{9}$$

i.e., the expected loss for a given decision rule $h\in\mathcal{H}$ and distribution $P(\cdot,\cdot)$ on the sample space $\mathcal{Z}=\mathcal{X}\times\mathcal{Y}$.

The estimation error arises because of the fact that the true joint distribution $P \in \mathcal{P}$ governing random behavior of the sampling data from the input and output product space, i.e., the sample space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ is unknown. It is logical to look at the risk due to estimation error from two aspects: a minimized empirical risk since for each candidate model, a random quantity evaluated from data and the average risk calculated from the limited number of training samples available.

The error in SML algorithm can be partitioned into approximation error and empirical error two parts:

$$\begin{aligned}
&\text{error}(h) = \sum_{x \in h \Delta c} D(x) \\
&= \text{error}_D(h) + \text{error}_P(h) \\
&= \underbrace{P\left\{h(x) \neq c(x) \mid x \overset{d}{\sim} D\right\}}_{\text{approximation error}} + \underbrace{\frac{1}{m}\sum_{i=1}^{m} \vartheta_{\{h(x)\neq c(x)\}}(x_i)}_{\text{estimation error or empirical error}}
\end{aligned} \tag{10}$$

where $\Delta$ is set symmetric difference operator.

The approximation error measures how well the functions in the chosen model space can approximate the underlying relationship between the output space and the input space, and in general improves as the "size" of our model space increases.

For better quantification of the risk, Haussler introduced a relative accuracy-wise metric in SML algorithm, see Hsu and Lin (2002), Jin et al. (2009).

**Definition 8:** The relative accuracy-wise metric $d_v : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ is defined by

$$d_v = \frac{|r-s|}{v+r+s} \tag{11}$$

where the parameter $v \in (0, \infty)$.

**Proposition 9:** The relative accuracy-wise metric $d_v, v > 0$ satisfies following three properties:

(1) For all non-negative reals *r* and *s*, $0 \le d_v(\cdot,\cdot) < 1$;

(2) For all non-negative reals *r* and *s*, $r \le s \le t$, then

$$\begin{aligned} d_v(r,s) &\le d_v(r,t), \\ d_v(r,t) &\le d_v(s,t) \end{aligned} \tag{12}$$

(3) For all non-negative reals *r* and *s*, $0 \le r \le s \le M$,

$$\frac{|r-s|}{v+2M} \le d_v(r,s) \le \frac{|r-s|}{v} \tag{13}$$

**Definition 10:** Let $r^*(P) = \inf_{h\in\mathcal{H}} r_h(P)$ and $r = r_{\hat{h}}(P)$ where $\hat{h}$ is the decision rule over $\mathcal{H}$ space such that it comes to "close" to minimizing the risk $r_{h,L}(P)$ over $\mathcal{H}$ and let $v > 0$, then the "close" minimizing metric is

$$d_v^*(P) = \frac{\left|r(P) - r^*(P)\right|}{v + r(P) + r^*(P)} \tag{14}$$

while the true approximating metric is

$$d_v^{h,L}(P) = \frac{\left|r(P) - r_{h,L}(P)\right|}{v + r(P) + r_{h,L}(P)} \tag{15}$$

The fundamental problem in Statistical Machine Learning (SML) is the statistical model selection, which is a tradeoff decision problem embedded intrinsically. Since what is the true functional form and how it relates to other explanatory variables are unknown to the statistical machine learning algorithm hence the hypothesized model family choice and the distributional law governing the sampling space is critical. If modeling family is too complex, the training data may be over-fitted (which causes an estimation error) and if a modeling family is too simple, it may result in a bad approximation of the function that we are trying to estimate (which causes an approximation error).

Furthermore, any statistical analysis and modeling including hypothesis inference and confidence interval itself is an optimization with respect to the given or pre-specified criterion subject to some constraints over the collected data. Mathematically speaking, statistical modeling is to search the optimal solution of an objective function subject to a set of constraints over the data. For example, the objective function of a classical regression model is a quadratic form subject to parameter constraints over the data, i.e., a least square optimization. While whether the SML algorithm is feasible and efficient, the optimization scheme plays the extremely important role.

In certain sense, whether or not a SML algorithm can be accepted and commercialized is dependent upon whether the response to the application environments of the SML product is sensitive, quick and accurate. Today's technical advancements in nano-industries do facilitate the best computer hardware, but the software developments are far behind the pace of hardware. Although more and more advanced computation techniques, such as vector machine, parallel computation etc, nevertheless, the optimization scheme is still the "basic" cell of all optimization approaches. The evolutionary algorithm, especially, GA scheme is a representative, is widely applied in SML. Just like genetic engineering, to seek a perfect body system, it is necessary to start with genetic cell improvements.

Therefore, we intend to discuss grey differential equation modeling family and our newly developed simple but efficient optimization scheme, λ-global optimization algorithm scheme (λ-scheme in short) as our contribution to SML algorithm in the current paper.

### 3.7.4 Grey Differential Equation Models and DEAR Formation

Grey differential equation model is proposed first by Professor Deng (1984, 1985, 2002, 2004). The model is quickly recognized and applied into many fields, see Lin and Lin (2006), Wen (2004), and Guo (2005, 2007), which successfully applied the grey differential equation approach into repairable system reliability analysis. In this section, it is necessary to explain the mathematical feature of the grey differential equation and the reason why we propose the DEAR modeling family to brings the grey differential equation into its mathematically rigorous status.

Without loss of generality, we will start with the one-variable first-order differential equation model (abbreviated as GM(1,1) model). The success of GM(1,1) model lies on the following two aspects: data AGO treatments (AGO is the abbreviation of accumulative generation operator) and a simple regression model coupled with a (whitening) differential equation model. Let $X^{(0)}=(x^{(0)}(1), x^{(0)}(2),\ldots, x^{(0)}(n))$ be a data sequence, and,

$$x^{(1)}(k)=\sum_{i-1}^{k}x^{(0)}(i),\quad k=2,3,4,\cdots,n \tag{16}$$

and

$$z^{(1)}(k)=\frac{1}{2}\left[x^{(1)}(k)+x^{(1)}(k-1)\right] \tag{17}$$

It is obvious that both $x^{(1)}(k)$ and $z^{(1)}(k)$ are approximation to primitive function $x(t)$ at $t=k.$ $x^{(1)}(k)$ is an accumulated sum and $z^{(1)}(k)$ is a further smoothed result (average) and believed an better approximation to the primitive function $x(t)$ at $t=k.$

**Definition 11:** Given a discrete positive real-valued data sequence $X^{(0)}=(x^{(0)}(1), x^{(0)}(2),\ldots, x^{(0)}(n))$ equation,

$$x^{(0)}(k)+\beta z^{(1)}(k)=\alpha,\quad k=2,3,4,\cdots,n \tag{18}$$

"coupled" with the first-order constant coefficient linear ordinary differential equation.

$$\frac{dx^{(1)}(t)}{dt} + \beta x^{(1)}(t) = \alpha \tag{19}$$

is called a GM(1,1) model with respect to a strictly positive discrete data sequence $X^{(0)} = (x^{(0)}(1), x^{(0)}(2), \ldots, x^{(0)}(n))$ . Parameter β is called the developing coefficient, parameter $\alpha$ is the grey input, term $x^{(0)}$ is called a *grey derivative* and term $x^{(1)}(k)$ is called the $k^{\text{th}}$ 1-AGO of $X^{(0)}$ value. Furthermore, the differential equation $dx^{(1)}/dt + \beta x^{(1)} = \alpha$ in Eq. (19) is called a whitening differential equation or a *shadow* equation of the grey differential equation, i.e., Deng (1984, 1985, 2002, 2004). Guo et al. (2006) named the differential equation Eq. (19) as the *associated differential equation* and also name the grey differential equation model in Eq. (18) as the *coupled regression*.

The unknown parameter values $(\alpha,\beta)$ can be determined in terms of a standard regression. Note that Eq. (18) can be written formally as in a simple regression theory,

$$y_k = \alpha + \beta x_k + \varepsilon_k, \quad k = 2,3,\cdots,n \tag{20}$$

where,

$$y_k = x^{0}\ \ k\ \ , x_k = -z^{1}\ \ k\ \ ,\ k = 2,3,\cdots,n \tag{21}$$

The estimate for regression parameter pair $(\alpha,\beta)$, denoted as $(a,b)$, can be calculated by,

$$(a,b)^T = (X^T X)^{-1} X^T Y \tag{22}$$

where,

$$X = \begin{bmatrix} 1 & -z^{(1)}(2) \\ 1 & -z^{(1)}(3) \\ \vdots & \vdots \\ 1 & -z^{(1)}(n) \end{bmatrix}, \quad Y = \begin{bmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \vdots \\ x^{(0)}(n) \end{bmatrix} \tag{23}$$

The grey filtering-prediction equation is:

$$\hat{x}^{(0)}(k) = \hat{x}^{(1)}(k) - \hat{x}^{(1)}(k\text{-}1) \tag{24}$$

where,

$$\hat{x}^{(1)}(k+1) = \left[ x^{(0)}(1) - \frac{a}{b} \right] e^{-bk} + \frac{a}{b} \tag{25}$$

Note that Eq. (25) is the discrete version of the solution to the differential equation Eq. (18):

$$x^{(1)} = \left[ x^{(1)}(0) - \frac{\alpha}{\beta} \right] e^{-\beta t} + \frac{\alpha}{\beta} \tag{26}$$

The typical goodness-of-fit measure of GM(1,1) model is the (absolute) relative error, i.e.

$$\hat{e}(k) = \frac{| x^{(0)}(k) - \hat{x}^{(0)}(k) |}{x^{(0)}(k)}, k = 2,3,4,\cdots,n \tag{27}$$

and the model efficiency is defined as:

$$\overline{e} = \frac{1}{n-1} \sum_{i=2}^{n} \hat{e}(i) \tag{28}$$

It should be aware that the prerequisite to build the standard GM(1,1) model proposed by Deng [17-21] is that the equal-gapped data sequence must be *strictly positive*.

From above review, it is quite evident that Professor Deng created a set of terminologies different from existing applied mathematical or statistical literature. For example, the AGO operator is nothing but a partial sum or a difference operation; the grey differential equation is merely a difference equation with derivative or integral approximation; the whitening differential equation or shadow differential is actually an associated differential equation. Those newly "invented" terminologies created a serious barrier for Western researchers to understand the grey differential equation idea. Furthermore, Deng's treatments on grey differential equation did not reveal adequately the mathematical features.

In order to keep the merits of the grey differential equation modelling the grey differential equation, Guo et al., (2007, 2008, 2009) proposed and developed DEAR (abbreviation of differential equation associated regression) modelling theory.

Guo et al. (2006) first reveals the fundamental link between appropriate pair of differential equation and regression in the form of difference equation in terms of the "Coupling Principle". In other words, the Coupling Principle plays a critical role for revealing the full mathematical picture about a grey differential equation and the corresponding differential equation. DEAR model formality is just stating such a fact in mathematical language. Table I, demonstrates how the Couple Principle acts by assuming a first-order differential equation and the corresponding first-order difference equation.

Table 3.7.1 Coupling Principle in the first-order DEAR Model

| Term | DE | REG |
|---|---|---|
| Discretization rule between DE model and Coupled regression model | | |
| Intrinsic feature | Continuous | Discrete |
| Explanatory variable and discrete collection | $t$ | $t_k,\ k=1,2,\cdots,n$ |
| 1st-order derivative and the 1st-order difference | $\frac{dx}{dt}$ | $\hat{x}^{(0)}(t_k)=\frac{\Delta x^{(1)}(t_k)}{\Delta t_k}$ $=\frac{x^{(1)}(t_k)-x^{(1)}(t_{k-1})}{t_k-t_{k-1}}$ |
| Primitive function and discretization | $x(t)$ | $x(t_k)$ |
| DE model and corresponding (central) difference equation | $\frac{dx(t)}{dt}=\alpha+\beta x(t)$ | $x^{(0)}(t_k)=\alpha+\beta\hat{x}(t_k)+\varepsilon_k$ or $\frac{\Delta x(t_k)}{\Delta t_k}=\alpha+\beta x(t_k)+\varepsilon_k$ |

Note that the difference has three versions: forward difference, backward difference and central difference. Without loss of generality, let us utilize a simple linear differential equation, $dx/dt=\alpha+\beta x$ in this paper for illustrative purpose. Let $\hat{x}_i^{1}$ denote an approximation to the primitive function $x(t)$ at $t_i$, and let $\Delta x_i/\Delta t_i$ be an approximation to the derivative function $dx/dt$ at $t_i$, with $\Delta x_i = x(t_i) - x(t_{i-1})$, $\Delta t_i = t_i - t_{i-1}$.

**Definition 12:** If a dynamic system governed by $dx/dt=\alpha+\beta x$ is taken $n$ observations at its first-order derivative level, denoted by $X^{0} = (x_1^{0}, x_2^{0}, \cdots, x_n^{0})$, the coupled equation system

$$\begin{cases} \dfrac{dx}{dt}=\alpha+\beta x \\ x_i^{0} = \alpha+\beta\hat{x}_i^{1}+\varepsilon_i,\ i=2,3,\cdots,n \end{cases} \tag{29}$$

is called Type I DEAR model.

**Definition 13:** If a dynamic system governed by Eq. (30) is sampled at its primitive function level with observation number $n$, denoted by $X^{1} = (x(t_1), x(t_2), \cdots, x(t_n))$, the coupled equation system

$$\begin{cases} \dfrac{dx}{dt}=\alpha+\beta x \\ \dfrac{\Delta x_i}{\Delta t_i} = \alpha+\beta x(t_i)+\varepsilon_i,\ i=2,3,\cdots,n \end{cases} \tag{30}$$

is called Type II DEAR model.

The second equations in Eq. (29) and Eq. (30) is called coupled regression, while the first ones is called the associated differential equation. The following three figures are used to illustrate the underlying mechanism behind DEAR modeling.

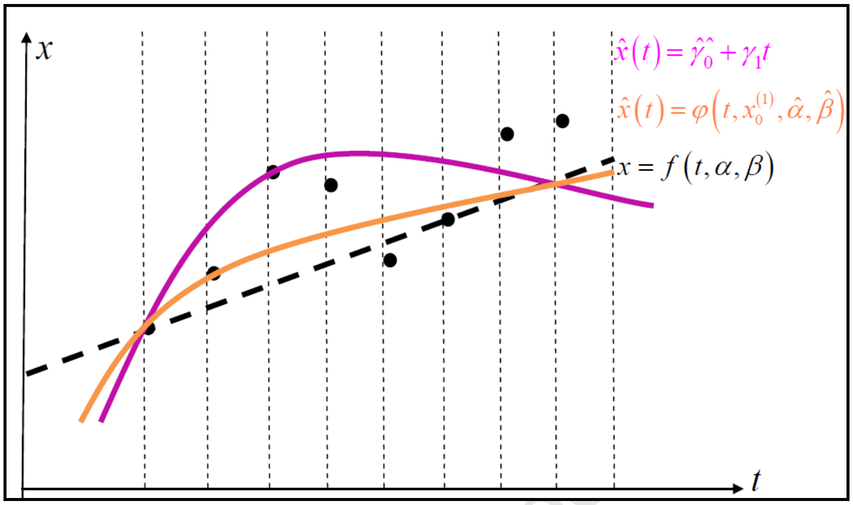



Figure 3.7.1 Two approximations to nonlinear curve $x\ t\ = f\ t;\alpha,\beta$ in $t,x$ space

Let us examine Type I DEAR model first. The system dynamics is governed by the linear differential equation $dx/dt=\alpha+\beta x$, or equivalently, nonlinear functional $x\ t\ = f\ t;\alpha,\beta$ . If the sample could be very large, it is possible to perform a non-linear statistical modeling in term of standard maximum likelihood procedure to estimate system parameter $\underline{\theta}=(\alpha,\beta)$. However, if only small amount observations are available, the "best" modeling exercise is to fit a simple regression model $\hat{x}(t)=\hat{\gamma}_0+\hat{\gamma}_1 t$ , called primitive regression, for approximating the system dynamics $x\ t\ = f\ t;\alpha,\beta$ . Figure 3.7.1 shows that the blue-dot straight line $\hat{x}(t)=\hat{\gamma}_0+\hat{\gamma}_1 t$ will poorly approximate nonlinear curve $x\ t\ = f\ t;\alpha,\beta$ in the $t,x$ space (or $t,x$ -coordinate system).

Let us consider the case where observations are collected at first-order derivative level, denoted as $X^0 = x_1^0, x_2^0, \cdots, x_n^0$ . By a linear transformation, approximations to primitive function level observations are obtained, denoted by $\{x(t_1),\hat{x}(t_2),\cdots,\hat{x}(t_n)\}$, say, by partial sum. In terms of Type I DEAR model thinking, we first fit the coupled regression, i.e., the second equation in DEAR equation system in Eq. (29) in the $x,x'$ space (or $x,x'$ -coordinate system), where $x'$ denotes the derivative of $x$ with respect to $t$, i.e., $x'=dx/dt$.

From the fitting of the coupled regression, $x_i^0 = \alpha + \beta\hat{x}_i^1 + \varepsilon_i$, the estimator of parameter $\underline{\theta} = (\alpha, \beta)$, denoted by $\hat{\underline{\theta}} = (\hat{\alpha}, \hat{\beta})$, is obtained. Now, in the $(x, x')$ space, we fit straight line $\hat{x}' = \hat{\alpha} + \hat{\beta}\hat{x}$ to approximate the straight line $x' = \alpha + \beta x$. It is obvious this model goodness-of-fit could be very good even with small amount observations.

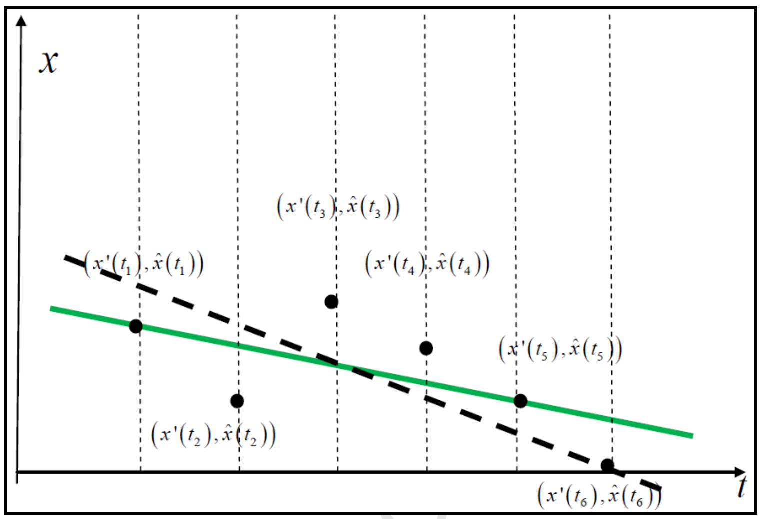



Figure 3.7.2 Type I approximation in $(x, x')$ space

Once the parameter $\underline{\theta} = (\alpha, \beta)$ is estimated, by solving the approximated linear differential equation $dx/dt = \hat{\alpha} + \hat{\beta}x$, we will obtain an approximated nonlinear curve $x' = \varphi(t; x_0^1, \hat{\alpha}, \hat{\beta})$, (yellow-colored curve in Figure 3.7.1), which is expected to approximate the primitive curve in relatively higher accuracy.

Let us consider the case in which the observations are collected at primitive function level, denoted as $X^1 = (x(t_1), x(t_2), \cdots, x(t_n))$. Then in terms of DEAR Type II model idea, the derivatives could be approximated, for example, by the divided difference, i.e., $\Delta x_i / \Delta t_i$, or other approaches available. Just as shown in Figure 3.7.3, fitting $\hat{x}' = \Delta x / \Delta t = \hat{\alpha} + \hat{\beta}x$ for approximating line $x' = \alpha + \beta x$. Similarly, the estimated parameter $\hat{\underline{\theta}} = (\hat{\alpha}, \hat{\beta})$ will lead the nonlinear approximation $x' = \varphi(t; x_0^1, \hat{\alpha}, \hat{\beta})$ to the primitive function $x(t) = f(t; \alpha, \beta)$ in $(t, x)$ space (shown in Figure 3.7.1).

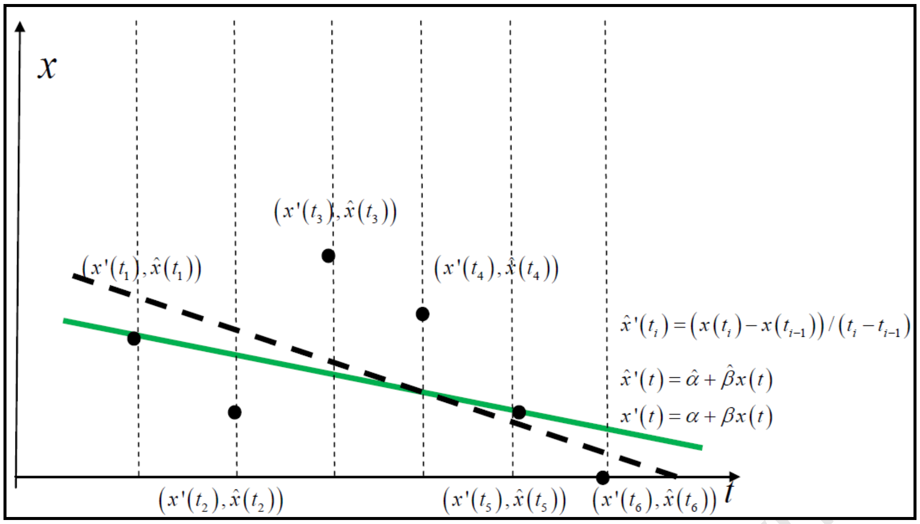



Figure 3.7.3 Type II approximation in $x, x'$ space

Furthermore, the solution to the associated differential equation (or the discretized solution) equipped with data-assimilated parameter estimates, is used for system analysis or prediction. We should emphasize here that the way a DEAR model uses system observations to solve the associated differential equation is different from that in common numerical algorithms for solving a differential equation numerically. In a DEAR model, we will obtain a closed-form functional solution (i.e., the primitive function) to the associated differential equation with optimal data-assimilated parameters. The availability of the closed-form primitive function, $x(t)$, will provide the great conveniences in the further investigation on the system behaviour under study. We acknowledge that the idea of obtaining a closed-form solution to the associated differential equation was suggested by the founder of Grey Systems, Deng (1984, 1985, 2002, 2004). The DEAR models defined in Definition 1 and 2 have a common feature that both of them start with (hypothesized) differential equation model and then the coupling regression model. Therefore, they are differential equation motivated regression (abbreviated as DEMR) models, Guo and Guo (2007).

We should emphasize that the DEAR modeling family includes very rich members and therefore DEAR modeling family will have potentially wide applications. Table 2 lists seven elementary models of the first-order DEAR.

Table 3.7.2 Seven Elementary Models in Type II DEAR Subfamily

| DEAR model | First-order DEAR model |
|---|---|
| 1 | $\begin{cases} \frac{dx}{dt} = \alpha_0 + \alpha_1 x \\ \frac{\Delta x(t_k)}{\Delta t_k} = \alpha_0 + \alpha_1 x(t_k) + \varepsilon_k \end{cases}$ |
| 2 | $\begin{cases} \frac{dx}{dt} = \alpha_0 e^{\delta t} + \alpha_1 x \\ \frac{\Delta x(t_k)}{\Delta t_k} = \alpha_0 e^{\delta t_k} + \alpha_1 x(t_k) + \varepsilon_k \end{cases}$ |
| 3 | $\begin{cases} \frac{dx}{dt} = \alpha_0 \sin(\omega t + \varpi) + \alpha_1 x \\ \frac{\Delta x(t_k)}{\Delta t_k} = \alpha_0 \sin(\omega t_k + \varpi) + \alpha_1 x(t_k) + \varepsilon_k \end{cases}$ |
| 4 | $\begin{cases} \frac{dx}{dt} = \alpha_0 e^{\delta t} \sin(\omega t + \varpi) + \alpha_1 x \\ \frac{\Delta x(t_k)}{\Delta t_k} = \alpha_0 e^{\delta t_k} \sin(\omega t_k + \varpi) + \alpha_1 x(t_k) + \varepsilon_k \end{cases}$ |
| 5* | $\begin{cases} \frac{dx}{dt} = \alpha_0 p_q(t) + \alpha_1 x \\ \frac{\Delta x(t_k)}{\Delta t_k} = \alpha_0 p_q(t_k) + \alpha_1 x(t_k) + \varepsilon_k \end{cases}$ |
| 6* | $\begin{cases} \frac{dx}{dt} = \alpha_0 e^{\delta t} p_q(t) + \alpha_1 x \\ \frac{\Delta x(t_k)}{\Delta t_k} = \alpha_0 e^{\delta t_k} p_q(t_k) + \alpha_1 x(t_k) + \varepsilon_k \end{cases}$ |
| 7* | $\begin{cases} \frac{dx}{dt} = \alpha_0 p_q(t) \sin(\omega t + \varpi) + \alpha_1 x \\ \frac{\Delta x(t_k)}{\Delta t_k} = \alpha_0 p_q(t_k) \sin(\omega t_k + \varpi) + \alpha_1 x(t_k) + \varepsilon_k \end{cases}$ |

Note: (*) involves a $q^{th}$-order polynomial function: $p_q(t) = p_0 + p_1 t + \cdots + p_q t^q$ $(q > 1)$

**Remark 14:** Table 2 lists seven often seen DEAR models with first-order differential equations. Similar DEAR modeling families can be also derived, see Guo and Guo (2009). As we mentioned early, DEAR models are developed in random fuzzy regression formation initially, hence the chance measure or average chance measure has to be involved for error structure analysis. It is in general true that the hybrid measure theory too complicated to be accepted by engineers or data analysts. We are fully aware that many forms of uncertainty, say, interval uncertainty, see Moore (1966), fuzzy uncertainty, see Zadeh (1978), grey uncertainty, see Deng (2002, 2004) and Liu and Lin (2006), and Liu's uncertainty (2007, 2010). Therefore, creating probabilistic DEAR model formulation and analysis is an urgent task.

## 3.7.5 Gaussian Process DEAR Modeling

Gaussian process regression (Rasmussen and Williams, 2006; Snelson, 2004; Williams and Rasmussen, 1996) is a Bayesian regression model (Bernarido and Smith, 1994; Billard and Diday, 2000; Chen, 1981; Lee, 1989) which take the intrinsic variance-covariance of a Gaussian process as the prior. It presumes a covariance function

$$v(x,x') = \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right) \tag{31}$$

where the maximum allowable variance is $\sigma_f^2$. As $v(x,x')$ approaches $\sigma_f^2$, which implies $f(x)$ is perfectly associated with $f(x')$. Multivariate versions of Eq. (31) can be constructed using quadratic forms such as

$$v(\underline{x},\underline{x}') = \sigma_f^2 \exp\left(-\frac{(\underline{x}-\underline{x}')^t AA^t (\underline{x}-\underline{x}')}{2l^2}\right). \tag{32}$$

For the novel Gaussian process regression,

$$v(x,x') = \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right) + \sigma_n^2 \delta(x,x') \tag{33}$$

where $\delta(x,x')$ is the Kronecker delta function.

The Gaussian process regression takes a form

$$y = f(\underline{x}) + G_t \tag{34}$$

where $\{G_t, t \geq 0\}$ is a Gaussian process. With sampled observations $\underline{y}$, and single observation $y_*$ to be predicted on the basis of the data and model, then we note

$$\begin{bmatrix} \underline{y} \\ y_* \end{bmatrix} \overset{d}{\sim} N\left( \begin{bmatrix} \underline{\mu} \\ \mu_* \end{bmatrix}, \begin{bmatrix} V & V_*^t \\ V_* & V_{**} \end{bmatrix} \right) \tag{35}$$

where

$$
\begin{aligned}
V &= \begin{bmatrix} v(x_1,x_1) & v(x_1,x_2) & \cdots & v(x_1,x_n) \\ v(x_2,x_1) & v(x_2,x_1) & \cdots & v(x_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ v(x_n,x_1) & v(x_2,x_n) & \cdots & v(x_n,x_n) \end{bmatrix}, \\
V_* &= \begin{bmatrix} v(x_*,x_1) & v(x_*,x_2) & \cdots & v(x_*,x_n) \end{bmatrix}, \\
V_{**} &= \begin{bmatrix} v(x_*,x_*) \end{bmatrix}.
\end{aligned} \tag{36}
$$

The prediction $y_* \mid \underline{y}$ satisfies

$$y_* \mid \underline{y} \overset{d}{\sim} N\left(\mu_* + V_* V^{-1}\left(\underline{y} - \underline{\mu}\right), V_{**} - V_* V^{-1} V_*^t\right). \tag{37}$$

If we wish to predict a vector $\underline{y}_* \mid \underline{y}$ whether without or with the presence in the model of a further *p* explanatory variables for the vector of means, along with any input variables determining known variance-covariance matrix $V$, then

$$\underline{y}_2 \mid \underline{y}_1 \overset{d}{\sim} N\left(X_2\underline{\beta} + V_{21}V_{11}^{-1}\left(\underline{y}_1 - X_1\underline{\beta}\right), V_{22} - V_{21}V_{11}^{-1}V_{12}\right). \tag{38}$$

Again, we use the first-order models to facilitate the DEAR models. Second order models can also developed similarly.

**Definition 15:** If a dynamic system governed by $dx/dt = \alpha + \beta x$ is taken random sample with size *n* at its derivative level, denoted by $X^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, \cdots, x_n^{(0)}\right)$, the coupled equation system

$$\begin{cases} \dfrac{dx}{dt} = \alpha + \beta x \\ x_i^{(0)} = \alpha + \beta \hat{x}_i^{(1)} + \sigma G_i, \ i = 2,3,\cdots,n \end{cases} \tag{39}$$

is the Type I Gaussian process DEAR model, where $G_i$ are the members of a Gaussian process indexed by integer set $\{2,3,\cdots,n\}$.

**Definition 16:** If a dynamic system governed by Eq. (40) is observed at its primitive level with observation number *n*, denoted by $X^{(1)} = \left(x(t_1), x(t_2), \cdots, x(t_n)\right)$, the coupled equation system

$$\begin{cases} \dfrac{dx}{dt} = \alpha + \beta x \\ \dfrac{\Delta x_i}{\Delta t_i} = \alpha + \beta x(t_i) + \sigma G_i, \ i = 2,3,\cdots,n \end{cases} \tag{40}$$

is called a Type II Gaussian process DEAR model, where $G_i$ are the members of a Gaussian process indexed by integer set $\{2,3,\cdots,n\}$.

**Remark 17:** Similar to random fuzzy DEAR models, the error terms are not necessarily taking integer indices, rather, any members from index set $[0,+\infty)$, for example, $\{t_0,t_1,\cdots,t_n\}\in\mathbb{R}^+$. In other words, the error terms may take members $\{G_{t_0},G_{t_2},\cdots,G_{t_n}\}$ from a Gaussian process $\{G_t,t\geq 0\}$. Then the Type I DEAR model takes the form

$$\begin{cases} \dfrac{dx}{dt}=\alpha+\beta x\ \ t \\ x_{t_i}^{\ 0}\ =\alpha+\beta\hat{x}_{t_i}^{\ 1}\ +\sigma G_{t_i},\ i=2,3,\cdots,n \end{cases} \tag{41}$$

and the Type II DEAR model takes the form

$$\begin{cases} \dfrac{dx}{dt}=\alpha+\beta x\ \ t \\ \dfrac{\Delta x_{t_i}}{\Delta t_i}=\alpha+\beta x\ \ t_i\ \ +\sigma G_{t_i},\ i=2,3,\cdots,n \end{cases} \tag{42}$$

where

$$\hat{x}^{\,1}\,(t_i)=\sum\nolimits_{i=1}^{i}\hat{x}^{\,0}\ \ t_i\ \ t_i-t_{i-1}\ \approx\int_{t_1}^{t_k}\left(\frac{dx}{dt}\right)dt \tag{43}$$

and

$$\Delta x_{t_i}/\Delta t_i=\left(x_{t_i}-x_{t_{i-1}}\right)/\left(t_i-t_{i-1}\right) \tag{44}$$

Note that the regression model is now a weighted regression model. However, as we state that the weight matrix $\Sigma$ is intrinsic to the Gaussian process $\{G_t,t\geq 0\}$.

**Definition 18:** Let $\Delta_h[f]=f(x+h)-f(x)$ be the forward difference, then the Newton's difference quotient is defined by $\Delta_h[f]/h$.

**Lemma 19:** Let $f(x)$ be a differentiable function with continuous derivative $f^{'}(x)$ at $x$. Then the forward Newton difference quotient approximation is

$$f^{'}(x)=\frac{f(x+h)-f(x)}{h}+\varepsilon_a \tag{45}$$

where the error term

$$\mathrm{E}[\varepsilon_a]=O\left(h^{-1}\right),V[\varepsilon_a]=O\left(h^{-2}\right) \tag{46}$$

The proof of the lemma is simply application of Taylor's expansion of a function.

**Proposition 20:** If we have observations on response function with random error $\varepsilon$, denoted as $\{f(x_1), f(x_2), \cdots, f(x_n)\}$, where $x_1 < x_2 < \cdots < x_n$, Then the variance of error vector is a diagonal matrix

$$V[\underline{\varepsilon}_a] = \sigma_a^2 \operatorname{diag}\left(h_i^{-2}\right). \tag{47}$$

Without loss of the generality, we use the simple Type II DEAR model for developing Gaussian process DEAR Type II model.

Let $\{G_t, t \ge 0\}$ be a general Gaussian process including the two processes as the special family members: the standard Brownian motion process $\{B_t, t \ge 0\}$ and standard fractional Brownian motion process $\left\{B_t^H, t \ge 0\right\}$, where Hurst index $H \in (0,1)$.

**Definition 21:** The coupled equation system

$$\begin{cases} \dfrac{df\ \ x}{dx} = \alpha + \beta f\ \ x \\ \dfrac{\Delta_{\Delta x_i}\left[\,f\ \ x_i\,\right]}{\Delta x_i} = \alpha + \beta f\ \ x_i\ \ + G_i,\ i = 2, 3, \cdots, n \end{cases} \tag{48}$$

where the error vector  has a variance-covariance matrix

$$V[\underline{\varepsilon}] = \sigma_a^2 \operatorname{diag}\left(h_i^{-2}\right) + \sigma_G^2 V. \tag{49}$$

with

$$V[\underline{\varepsilon}_G] = \begin{cases} \sigma_B^2 \begin{bmatrix} x_2 & x_2 & \cdots & x_2 \\ x_2 & x_3 & \cdots & x_3 \\ \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & \cdots & x_n \end{bmatrix} & \text{if } G_t = B_t \\ \sigma_B^2 \begin{bmatrix} c_{22} & c_{23} & \cdots & c_{2n} \\ c_{32} & c_{33} & \cdots & c_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n2} & c_{n3} & \cdots & c_{nn} \end{bmatrix} & \text{if } G_t = B_t^H \end{cases} \tag{50}$$

$$c_{ij} = \frac{1}{2}\left(x_i^{2H} + x_j^{2H} - \left|x_i - x_j\right|^{2H}\right),\ i, j = 2, 3, \cdots, n$$

by noticing that

$$\mathrm{E}[G_s G_t] = \begin{cases} s \wedge t & \text{if } G_t = B_t \\ \dfrac{1}{2}\left(s^{2H} + t^{2H} - |s-t|^{2H}\right) & \text{if } G_t = B_t^H \end{cases} \tag{51}$$

Therefore, the likelihood function takes a form

$$l(\underline{\theta} \mid \underline{x}) \propto \exp\left(-\frac{1}{2l^2}(\underline{y} - X\underline{\theta})^t \left(\sigma_a^2 \text{diag}\left(h_i^{-2}\right)\right)^{-1} (\underline{y} - X\underline{\theta})\right) \tag{52}$$

where

$$\begin{aligned} \underline{\theta}^t &= (\alpha, \beta), \\ X &= \begin{bmatrix} 1 & f(x_2) \\ 1 & f(x_3) \\ \vdots & \vdots \\ 1 & f(x_n) \end{bmatrix}, \\ \underline{y}^t &= \left(\frac{\Delta_{\Delta_{x_2}}[f]}{\Delta_{x_2}}, \frac{\Delta_{\Delta_{x_3}}[f]}{\Delta_{x_3}}, \cdots, \frac{\Delta_{\Delta_{x_n}}[f]}{\Delta_{x_n}}\right) \end{aligned} \tag{53}$$

Based on the likelihood expression, we can easily to derive the joint distribution for $\underline{\theta}$ given the Gaussian process prior

$$f(\underline{\theta}, \underline{x}) \propto \exp\left(-\frac{1}{2l^2}(\underline{y} - X\underline{\theta})^t \left(\sigma_a^2 \text{diag}\left(h_i^{-2}\right) + \sigma_G^2 V\right)^{-1} (\underline{y} - X\underline{\theta})\right) \tag{54}$$

and the posterior for $\underline{\theta}$ is

$$f(\underline{\theta} \mid \underline{x}) \propto \frac{\exp\left(-\frac{1}{2l^2}(\underline{y} - X\underline{\theta})^t \left(\sigma_a^2 \text{diag}\left(h_i^{-2}\right) + \sigma_G^2 V\right)^{-1} (\underline{y} - X\underline{\theta})\right)}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \exp\left(-\frac{1}{2l^2}(\underline{y} - X\underline{\theta})^t \left(\sigma_a^2 \text{diag}\left(h_i^{-2}\right) + \sigma_G^2 V\right)^{-1} (\underline{y} - X\underline{\theta})\right) d\alpha\, d\beta} \tag{55}$$

As to the Bayes estimator of $\underline{\theta}$, denoted by $\underline{\tilde{\theta}}^t = (\tilde{\alpha}, \tilde{\beta})$ , we can obtain them from $f(\alpha, \beta \mid \underline{x})$.

We are going to use the freestyle 100 meters swimming test records (Example 6.1, in Deng [19]) as an illustrative example to expose the model fitting procedure. The test records are listed in Table 3.



Table 3.7.3 100m freestyle swimming records (in second)

| $k$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $x(k)$ | 58.9 | 59.1 | 59.3 | 59.5 | 59.7 | 59.55 | 59.4 | 59.3 |

The observations are taken at its primitive level, the time consumed for completing 100 meters in freestyle swimming. Therefore, Type II DEAR modeling may be engaged.

Table 3.7.4 The first-order differences

| $k$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $x^{(0)}(k)$ | | 0.2 | 0.2 | 0.2 | 0.2 | -0.15 | -0.15 | -0.1 |

Because the observations show a decreasing and increasing pattern, the hypothesized differential equation may take the form:

$$\frac{dx}{dt} + \beta x = \left(\alpha_0 + \alpha_1 t + \alpha_2 t^2\right) e^{\delta t} \tag{56}$$

Note that the solution to Eq. (56) is $x = x_h + x_p$, where $x_h = c_0 e^{-\beta t}$, is the solution to the homogeneous equation, and $x_p = \left(A_0 + A_1 t + A_2 t^2\right) e^{\delta t}$ is the particular solution satisfying:

$$\begin{aligned} &\frac{dx_p}{dt} + \beta x_p = \left(A_1 + 2A_2 t\right) e^{\delta t} \\ &+\delta\left(A_0 + A_1 t + A_2 t^2\right) e^{\delta t} \\ &+\beta\left(A_0 + A_1 t + A_2 t^2\right) e^{\delta t} \\ &= \left(\alpha_0 + \alpha_1 t + \alpha_2 t^2\right) e^{\delta t} \end{aligned} \tag{57}$$

which results in the equation system:

$$\begin{cases} \alpha_0 = A_1 + \left(\beta + \delta\right) A_0 \\ \alpha_1 = 2A_2 + \left(\beta + \delta\right) A_1 \\ \alpha_2 = \left(\beta + \delta\right) A_2 \end{cases} \tag{58}$$

and hence

$$\begin{cases} A_0 = \dfrac{\left(\beta+\delta\right)^2 \alpha_0 - \left(\beta+\delta\right)\alpha_1 + 2\alpha_2}{\left(\beta+\delta\right)^2} \\ A_1 = \dfrac{\left(\beta+\delta\right)\alpha_1 - 2\alpha_2}{\left(\beta+\delta\right)^2} \\ A_2 = \dfrac{\alpha_2}{\beta+\delta} \end{cases} \tag{59}$$

Therefore, the coupled regression takes the form:

$$\begin{aligned} x^{(0)}(k) &= \alpha_0 e^{\delta k} + \alpha_1 e^{\delta k} k + \alpha_2 e^{\delta k} k^2 \\ &\quad + \alpha_3 \left(-x^{(1)}(k)\right) + \sigma G_k \\ k &= 2, 3, \cdots, 8 \end{aligned} \tag{60}$$

Summarizing the above arguments, we have probabilistic DEAR model:

$$\begin{cases} \dfrac{dx}{dt} + \beta x = \;\; \alpha_0 + \alpha_1 t + \alpha_2 t^2 \;\; e^{\delta t} \\ x^{(0)}(k) = \alpha_0 e^{\delta k} + \alpha_1 e^{\delta k} k + \alpha_2 e^{\delta k} k^2 + \;\; \beta \;\; -x^{(1)}(k) \;\; + \sigma G_k \end{cases} \tag{61}$$

Then the weighted regression model is

$$\begin{aligned} y_k &= \alpha_0 e^{\delta k} + \alpha_1 e^{\delta k} k + \alpha_2 e^{\delta k} k^2 \\ &\quad + \alpha_3 \ -x^{(1)}(k) \ + \sigma G_k, \\ k &= 2,3,\cdots,8 \end{aligned} \tag{62}$$

with

$$\begin{aligned} &\underline{y} = \begin{bmatrix} 0.20 \\ 0.20 \\ \vdots \\ -0.10 \end{bmatrix}, \underline{G} = \begin{bmatrix} \sigma G_2 \\ \sigma G_3 \\ \vdots \\ \sigma G_8 \end{bmatrix}, \underline{\alpha} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}, \\ &X = \begin{bmatrix} e^{2\tilde{\delta}} & 2e^{2\tilde{\delta}} & 4e^{2\tilde{\delta}} & -59.1 \\ e^{3\tilde{\delta}} & 3e^{3\tilde{\delta}} & 9e^{3\tilde{\delta}} & -59.3 \\ \vdots & \vdots & \vdots & \vdots \\ e^{8\tilde{\delta}} & 8e^{8\tilde{\delta}} & 64e^{8\tilde{\delta}} & -59.3 \end{bmatrix} \end{aligned} \tag{63}$$

Denote $\beta = \alpha_3$, set up the initial $\tilde{\delta} = 1.0$. Actually, the objective function is

$$l(\underline{\alpha},\delta) = \left(\underline{y} - X_\delta \underline{\alpha}\right)' \Sigma^{-1} \left(\underline{y} - X_\delta \underline{\alpha}\right) \tag{64}$$

By minimizing the objective function $l(\underline{\alpha},\delta)$ in terms of λ-global optimization scheme, we get an optimal solution $\left(\hat{\underline{\alpha}},\hat{\delta}\right)$. Then utilizing Matlab weighted least regression function for the fixed $\hat{\delta} = 4.3370$ as given and assuming Brownian motion process, refit

$$\underline{y} = X_{\hat{\delta}} \underline{\alpha} + \underline{B} \tag{65}$$

Keep in mind, $k = 2,3,\cdots,8$.

The refitted weighted regression model is

$$\begin{aligned} y_k &= \Big( \underset{(1.17E\text{-}12)}{-4.96E\text{-}11} + \underset{3.13E\text{-}13}{1.33E\text{-}11} \times k - \underset{2.08E\text{-}14}{8.83E\text{-}13} \times k^2 \Big) e^{4.3370*k} \\ &\quad - \underset{8.17E\text{-}05}{0.00338339} \ -x^{(1)}(k) \ , \\ k &= 2,3,\cdots,8 \end{aligned} \tag{66}$$

The model average accuracy (i.e., average relative error) is 4.7%. Table 5 lists the model fitting details.

Table 3.7.5 The fitted Values and Error & Relative Errors

| $k$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| $x(k)$ | 58.9 | 59.1 | 59.3 | 59.5 | 59.7 | 59.55 | 59.4 | 59.3 |
| $x^{(0)}(k)$ | | 0.2 | 0.2 | 0.2 | 0.2 | -0.15 | -0.15 | -0.1 |
| $\hat{x}^{(0)}(k)$ | | 0.2 | 0.201 | 0.201 | 0.188 | -0.161 | -0.161 | -0.111 |
| Relative error | | 0% | 0.03% | 0.05% | 6.02% | 7.47% | 7.45% | 11.1% |

A final comment on DEAR modeling of the data is it is possible to fit other DEAR models, say, with sine terms for catching the down-up trend. It is also to emphasize that due to the features of observational data $\{x(k), k=1,2,\cdots,8\}$, the DEAR model in Eq.(63) is seeking the best goodness-of-fit not for predictions, but other appropriate DEAR model could be better for prediction. Also, the searching optimal solution requires more efficient and fast global optimization scheme. Therefore, we switch our attention into the λ-global optimization scheme, which is actually an organic part or a companion of DEAR modelling developments.

### 3.7.6 λ-Global Optimization Scheme

During the repairable system DEAR modelling exercises, it was often finding that the optimization algorithm's performances, say, GA in Matlab or Ant Colony algorithm were not satisfactory, which stimulated our immediate interest: develop a optimization scheme can cope with DEAR modelling processes.

The λ-global optimization scheme or λ-scheme in short is created by imitating an ancient human body system, Cui et al. (2009, 2010). In its searching scheme, except the necessary mathematical computations for evaluating the objective function and the creation of the initial "searching population" randomly, the scheme only involves if-else logical operation and sort procedure. In contrast to existing global optimization algorithms, particularly genetic algorithm (abbreviated as GA), the λ-scheme engages the simplest mathematics but reaches the highest searching efficiency. In certain circumstances searching speed by the λ-scheme is faster than that of GA. Even in some case, there is no solution by GA searching, but the λ-scheme searching does. Therefore, it is very beneficial to bring λ-global optimization scheme into statistical machine learning algorithm. Let us now introduce the basic concepts of the λ-scheme.

**Definition 22:** A string in an algorithm, denoted by $\underline{e} = e_1 e_2 \cdots e_l$, is a sequence of $l$ elements from the membership set Θ. The total number of the elements, $l$, composed of the string $\underline{e}$ is called the length of the string.

**Definition 23:** In a string algorithm, in order to represent an $N$-dimensional Euclidean point $\underline{x}' = (x_1, x_2, \cdots, x_N)$, a string $\underline{e}$ is typically constituted by $N$ segmental strings, whose length are $u$, i.e., the string $\underline{e} = \underline{e}_1\underline{e}_2\cdots\underline{e}_N$ with length $l = Nu$, where the $n^{th}$ segment of the string, or the $n^{th}$ segmental string, $\underline{e}_n = e_{u(n-1)+1}e_{u(n-1)+2}\cdots e_{nu}$, is of length $u$, ($n=1,2,\cdots,N$).

**Definition 24:** A triple $(v,u,N)$ is called string configuration, where $v =$ total number of elements in the membership set Θ, $u =$ the length of segmental string $\underline{e}_n = e_{u(n-1)+1}e_{u(n-1)+2}\cdots e_{nu}$, and $N =$ total number of segmental strings composing of the string $\underline{e} = \underline{e}_1\underline{e}_2\cdots\underline{e}_N$. Let $\mathbb{S} = \{\underline{e} : e \in \Theta\}$ denote the string space generated from $v$-element membership set $\Theta_v$ and $\mathbb{S}_{(v,u,N)} = \left\{\underline{e}_1\underline{e}_2\cdots\underline{e}_N : e_{(n,j_n)} \in \Theta_v\right\}$ the $(v,u,N)$ configuration string space on $\Theta_v$.

For example, we are searching the global optimum of a two-dimensional function, $f(X_1, X_2)$. Let

$D_1 \equiv [u_{1,\min}, u_{1,\max}]$ be the searching domain of $X_1$;

$D_2 \equiv [u_{2,\min}, u_{2,\max}]$ be the searching domain of $X_2$;

$u_{1,\min}, u_{1,\max}$ be the lower bound and upper bound of $X_1$;

$u_{2,\min}, u_{2,\max}$ be the lower bound and upper bound of $X_2$.

Then, Eq. (67) and Eq. (68) specify the relationship between $(X_1, X_2)$ and the (5, 6, 2) configuration string $\underline{e} = \underline{e}_1\underline{e}_2 = e_1e_2\cdots e_6e_7e_8\cdots e_{12}$:

$$X_1 = u_{1,\min} + (u_{1,\max} - u_{1,\min})\sum_{j=1}^{6} e_j \frac{5^{6-j}}{5^6} \tag{67}$$

and

$$X_2 = u_{2,\min} + (u_{2,\max} - u_{2,\min})\sum_{j=7}^{12} e_j \frac{5^{12-j}}{5^{12}} \tag{68}$$

where $e_j$, $j = 1,2,\cdots,12$ denote the elements taking numbers 0, 1, 2, 3, or 4.

**Definition 25:** An index pair $(n, j_n)$ indicates the $n^{th}$ segmental string and the $j^{th}$ element (position) in the $n^{th}$ segmental string.

Keep in mind that the subscript of $n^{\text{th}}$ segmental string, $u(n-1)+1, u(n-1)+2, \cdots, nu$ actually point to the element (or cell) position within the whole string, $\underline{e}$.

Once we setup values of $D_1 \equiv \left[u_{1,\min}, u_{1,\max}\right]$ and $D_2 \equiv \left[u_{2,\min}, u_{2,\max}\right]$, the fitness value of a objective function $f(x_1, x_2)$ fitness value is readily to calculate according to Eq. (67) and Eq. (68).

**Definition 26:** A string vector, denoted by $\overline{e^1}$, is a column vector taking strings as its components. The dimensionality indicates how many strings are used to construct a string vector. In other words, a string vector is a matrix constituted by string elements $e_{(i),(j)} \in \Theta = \{0,1,2,3,4\}$, where $i$ indicates the $i^{th}$ component string $\underline{e}_{(i)}$ in string vector $\overline{e^1}$.

It is obvious that for (5, 6, 2) configuration string vector, if we assume that the dimensionality $\overline{e^1}$ is 100, which represents as 100 pairs of variables $X_1$ and $X_2$. As matter of fact, a string vector $\overline{e^1} = \left(e_{(i),(j)}\right)_{M\times l}$ is a matrix of elements $e_{(i),(j)} \in \Theta$ with size $M \times l$.

$$\overline{e^1} = \begin{bmatrix} \underline{e}_{(1)} \\ \underline{e}_{(2)} \\ \vdots \\ \underline{e}_{(100)} \end{bmatrix} = \begin{bmatrix} e_{(1),(1)} & \cdots & e_{(1),(6)} & e_{(1),(7)} & \cdots & e_{(1),(12)} \\ e_{(2),(1)} & \cdots & e_{(2),(6)} & e_{(2),(7)} & \cdots & e_{(2),(12)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ e_{(100),(1)} & \cdots & e_{(100),(6)} & e_{(100),(7)} & \cdots & e_{(100),(12)} \end{bmatrix} \tag{69}$$

i.e., a matrix or string vector $\overline{e^1} = \left(e_{(i),(j)}\right)_{100\times 12}$.

Figure 3.7.4 intuitively shows a string vector of $\left(e_{(i),(j)}\right)_{100\times 12}$ and its cycle vectors.

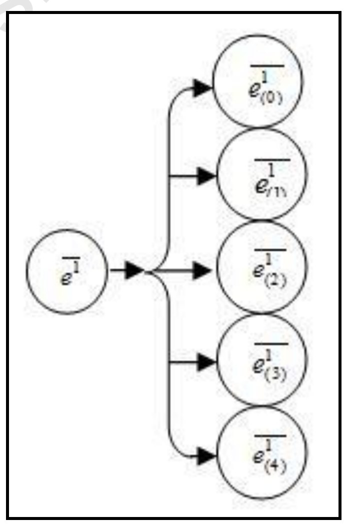



Figure 3.7.4 A string vector $\overline{e^1}$ and its cyclic vectors

**Definition 27:** λ comparison operation of the first kind means that the value of the element $e_{(i),(j)}$ follows $0 \to 1 \to 2 \to 3 \to 4$ criterion to change.

**Example 28:** Let $\underset{\sim}{e}^1_{(1)}, \underset{\sim}{e}^1_{(2)}, \cdots\cdots, \underset{\sim}{e}^1_{100}$ are component strings in string vector $\overline{e^1}$. Let $e^1_{(i),(1)}, e^1_{(i),(2)}, \cdots, e^1_{(i),(12)}$ are elements in $i^{\text{th}}$ component string $\underset{\sim}{e}^1_{(i)}$, respectively, $i = 1, 2, \cdots, 100$. Then

For i=1:1:100-2
For j=1:1:12
If $e^1_{(i),(j)} \equiv e^1_{(i+1),(j)}$
$e^1_{(i+1),(j)} \leftarrow \lambda\left[e^1_{(i),(j)}\right]$
ELSE IF $e^1_{(i),(j)} \equiv e^1_{(i+2),(j)}$
$e^1_{(i+2),(j)} \leftarrow \lambda\left[e^1_{(i),(j)}\right]$
END

**Definition 29:** $\lambda^{-1}$ comparison operation of the second kind means that the value of the element $e_{(i),(j)}$ follows $0 \to 4 \to 3 \to 2 \to 1$ criterion to change.

**Example 30:** Let, $\underset{\sim}{e}^1_{(1)}, \underset{\sim}{e}^1_{(2)}, \cdots\cdots, \underset{\sim}{e}^1_{100}$ are component strings in string vector $\overline{e^1}$. Let $e^1_{(i),(1)}, e^1_{(i),(2)}, \cdots, e^1_{(i),(12)}$ are elements in $i^{\text{th}}$ component string $\underset{\sim}{e}^1_{(i)}$, respectively, $i = 1, 2, \cdots, 100$. Then

For i=1:1:100-2
For j=1:1:12
If $e^1_{(i),(j)} \equiv e^1_{(i+1),(j)}$
$e^1_{(i+1),(j)} \leftarrow \lambda^{-1}\left[e^1_{(i),(j)}\right]$
ELSE IF $e^1_{(i),(j)} \equiv e^1_{(i+2),(j)}$
$e^1_{(i+2)(j)} \leftarrow \lambda^{-1}\left[e^1_{i(j)}\right]$
END



Furthermore, we state the assumption on the initial set of string vectors.

*Initialization Assumption***:** Let the initial string vector be $\overline{e^0} = \left(e^{(0)}_{(i),(j)}\right)_{M \times l}$ such all the elements in the $i^{th}$ component string $\underset{\sim}{e}^0_{(i)}$ are mutually independent, $i = 1, 2, \cdots, M$.

Let $D \equiv [u_{\min}, u_{\max}]^N$ be the searching domain for an objective function $f(\underset{\sim}{x})$ defined in *N*-dimensional Euclidean space $\mathbb{R}^N$. It is obvious *D* determines the scope of searching globally. Mathematically, the linear system linking the $i^{\text{th}}$ component string $\underset{\sim}{e}^1_{(i)}$, the system state $\underset{\sim}{x}_i$ can be expressed by

$$\begin{cases} x_{i,1} = u_{\min} + \left(u_{\max} - u_{\min}\right) \sum_{j=1}^{u} e_{(i),(j)} \frac{5^{u-j}}{5^{u}} \\ x_{i,2} = u_{\min} + \left(u_{\max} - u_{\min}\right) \sum_{j=u+1}^{2u} e_{(i),(j)} \frac{5^{2u-j}}{5^{2u}} \\ \vdots \\ x_{i,N} = u_{\min} + \left(u_{\max} - u_{\min}\right) \sum_{j=(n-1)u+1}^{nu} e_{(i),(j)} \frac{5^{Nu-j}}{5^{nu}} \end{cases} \tag{70}$$

Let the weight matrix be

$$O_{n\times nu} = \begin{bmatrix} \frac{5^{u-1}}{5^u} & \cdots & \frac{5^0}{5^u} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \frac{5^{u-1}}{5^u} & \cdots & \frac{5^0}{5^u} & 0 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \frac{5^{u-1}}{5^u} & \cdots & \frac{5^0}{5^u} \end{bmatrix} \tag{71}$$

and furthermore, let

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}; \ \underline{u}_{\min} = \begin{bmatrix} u_{\min} \\ u_{\min} \\ \vdots \\ u_{\min} \end{bmatrix}; \ u_r = u_{\max} - u_{\min} \tag{72}$$

and write string $\underline{e} = e_1 e_2 \cdots e_l$ in column vector form, (i.e., $nu \times 1$ column vector of element $e_i$), that is

$$\underline{e}^{'}_{Nu\times 1} = \begin{bmatrix} \underline{e}^{'}_{1} \\ \vdots \\ \underline{e}^{'}_{u} \\ \underline{e}^{'}_{u+1} \\ \vdots \\ \underline{e}^{'}_{2u} \\ \vdots \\ \underline{e}^{'}_{Nu-1} \\ \vdots \\ \underline{e}^{'}_{Nu} \end{bmatrix} \tag{73}$$

Then the candidate solution is a linear transformation of the (5,*u*,*N*) configured string representation

$$\underline{x} = \underline{u}_{\min} + u_r O_{n\times Nu} \underline{e}^{'}_{Nu} \tag{74}$$

**Definition 31:** Let $\underline{x}^{(q)} = \underline{u}_{\min} + u_r \lambda^{(q)} [\underline{e}] O', q \in \{0,1,2,3,4\}$, then,

$$\left\{ f\left(\underline{x}^{(0)}\right), f\left(\underline{x}^{(1)}\right), f\left(\underline{x}^{(2)}\right), f\left(\underline{x}^{(3)}\right), f\left(\underline{x}^{(4)}\right) \right\}_{\underline{e}} \tag{75}$$

is called the cyclic set of objective function values with respect to string $\underline{e}'_{Nu\times 1}$.

**Definition 32:** ($\lambda$ comparison operation on two component strings) If we compare two component strings $\underline{e}_{(1)}$ and $\underline{e}_{(2)}$. Assume that String $\underline{e}_{(1)}$ (candidate solution)'s fitness value is better than string $\underline{e}_{(2)}$'s, and then we managed some changes to $\underline{e}_{(2)}$.

Let $L$ be the length of component strings $\underline{e}_{(1)}$ and $\underline{e}_{(2)}$. Let $e_{(i),(j)}$ and $e_{(2),(j)}$ be one of the element of component strings $\underline{e}_{(1)}$ and $\underline{e}_{(2)}$ respectively, and $g_1(\cdot)$ and $g_2(\cdot)$ be two different kind of $\lambda$ comparison operation functions.

$$g_1\left(e_{(2),(j)}\right) = \begin{cases} \lambda\left\lfloor e_{(2),(j)} \right\rfloor, & \text{if } e_{(1),(j)} = e_{(2),(j)} \\ e_{(2),(j)}, & \text{if } e_{(1),(j)} \neq e_{(2),(j)} \end{cases} \tag{76}$$

and

$$g_2\left(e_{(2),(j)}\right) = \begin{cases} \lambda^{-1}\left[ e_{(2),(j)} \right], & \text{if } e_{(1),(j)} = e_{(2),(j)} \\ e_{(2),(j)}, & \text{if } e_{(1),(j)} \neq e_{(2),(j)} \end{cases} \tag{77}$$

Because $\lambda$ comparison operation have two different kinds of functions, and each function applies on only one single string vector $\underline{e}_{(i)}$, so we may only describes $g(\cdot)$ „s applications as a sample for other similar function.

**Definition 33:** Given a string vector $\left(e_{(i),(j)}\right)_{M\times Nu}$. A string $\underline{e}_{(i)} = e_{(i),(1)}e_{(i),(2)}\cdots e_{(i),(u)}e_{(i),(u+1)}\cdots e_{(i),(2u)}\cdots e_{(i),((N-1)u+1)}\cdots e_{(i),(Nu)}$ represents a candidate solution has $N$ components. By ranking of the fitness values from best to worst, we have a sorted string vector $\overline{e'}$, where $e_{ij}$ denotes any element in $\overline{e'}$ at $i^{th}$ row, $j^{th}$ column. Then $\lambda$ Comparison operation in strings vector defined as:

If $i \geq 3$

$$g\left(e_{ij}\right) = \begin{cases} \lambda\left[e_{ij}\right], & \text{if } e_{ij} = e_{(i-1),(j)} \neq e_{(i-2),(j)} \\ \lambda\left[e_{ij}\right], & \text{if } e_{ij} \neq e_{(i-1),(j)} = e_{(i-2),(j)} \\ \lambda^{(2)}\left[e_{ij}\right], & \text{if } e_{ij} = e_{(i-1),(j)} = e_{(i-2),(j)} \\ e_{ij}, & \text{if } e_{ij} \neq e_{(i-1),(j)} \neq e_{(i-2),(j)} \end{cases} \tag{78}$$

where

$$
\begin{aligned}
&p\left(\lambda\left[e_{ij}\right]\right)=p\left(e_{ij}=e_{(i-1)j}\neq e_{(i-2)j}\right)+p\left(e_{il}\neq e_{(i-1)j}=e_{(i-2)j}\right)\\
&=0.16+0.16=0.32;\\
&p\left(\lambda^{(2)}\left[e_{ij}\right]\right)=0.04,\, p\left(e_{ij}\right)=0.64
\end{aligned}
\tag{79}
$$

If $i=2$

$$
g\left(e_{ij}\right)=\begin{cases}\lambda\left[e_{ij}\right], & \text{if } e_{ij}=e_{(i-1),(j)}\\ e_{ij}, & \text{if } e_{ij}\neq e_{(i-1),(j)}\end{cases}
\tag{80}
$$

where

$$
p\left(\lambda\left[e_{ij}\right]\right)=0.2 \text{ and } p\left(e_{ij}\right)=0.8
\tag{81}
$$

If $i = 1$

$$
g\left(e_{ij}\right)=e_{ij}
\tag{82}
$$

Note that at each looping time $t$, a $\lambda$ comparison operation on whole string vector $\bar{e}$ will result in a new conditional random variable.

**Definition 34:** Given a matrix $\left(e_{(i),(j)}\right)_{M\times Nu}$ in column vector form, then the set of strings after an expansion

$$
\lambda^{\text{expansion}}\left[\underline{e}'\right]\triangleq\left\{\lambda^{(0)}\left[\underline{e}'\right],\lambda^{(1)}\left[\underline{e}'\right],\lambda^{(2)}\left[\underline{e}'\right],\lambda^{(3)}\left[\underline{e}'\right],\lambda^{(4)}\left[\underline{e}'\right]\right\}
\tag{83}
$$

is called the $\lambda$ expansion set.

Now we would like to examine the string state change in $\lambda$ expansion set after a $\lambda$ comparison operation executed in string vector.

If $i\geq 3$

$$
\lambda^{\text{expansion}}g\left(e_{ij}\right)=\begin{cases}\lambda^{(k)}\left[\underline{e}_{ij}\right]\lambda\left[e_{ij}\right], & \text{if } e_{ij}=e_{(i-1),(j)}\neq e_{(i-2),(j)}\\ \lambda^{(k)}\left[\underline{e}_{ij}\right]\lambda\left[e_{ij}\right], & \text{if } e_{ij}\neq e_{(i-1),(j)}=e_{(i-2),(j)}\\ \lambda^{(k)}\left[\underline{e}_{ij}\right]\lambda^{(2)}\left[e_{ij}\right], & \text{if } e_{ij}=e_{(i-1),(j)}=e_{(i-2),(j)}\\ \lambda^{(k)}\left[\underline{e}_{ij}\right], & \text{if } e_{ij}\neq e_{(i-1),(j)}\neq e_{(i-2),(j)}\end{cases}
\tag{84}
$$

If $i=2$

$$
\lambda^{\text{expansion}}\left(g\left(e_{ij}\right)\right)=\begin{cases}\lambda^{(k)}\left[\underline{e}_{ij}\right]\lambda\left[\underline{e}_{ij}\right], & \text{if } e_{ij}=e_{(i-1),(j)}\\ \lambda^{(k)}\left[\underline{e}_{ij}\right], & \text{if } e_{ij}\neq e_{(i-1),(j)}\end{cases}
\tag{85}
$$

If $i = 1$

$$\lambda^{\text{expansion}}\left(g\left(e_{ij}\right)\right)=\lambda^{(k)}\left[\underline{e}_{ij}\right] \tag{86}$$

where $k=0,1,2,3,4;\ i=1,2,\cdots,M;\ j=1,2,\cdots,Nu$ . $L=Nu$ is the length of string, $N$ represents dimensionality of strings in the string vector.

Both $\lambda$ expansion and $\lambda$ comparison operations in string vector are taken after ranking the string vector according to the value of objective function $f(\underline{x})$. After ranking, the fitness values corresponding to strings $\underline{e}_{(i)},\underline{e}_{(i-1)},\underline{e}_{(i-2)}$ are supposed to be very close to those corresponding to whole vector strings.

Therefore, what we need to find out are whether or not some same elements exist in each of $\underline{e}_{(i)},\underline{e}_{(i-1)},\underline{e}_{(i-2)}$ (three strings) to ensure those repeated elements in the strings are the reason why the fitness values are similar.

Consequently, we can select only one string from rejoined 5 states $\lambda^{(k)}\left\lfloor \underline{e_i^{new}} \right\rfloor$, *k=0,1,2,3,4* of strings.

After carrying on the above process recursively, the sequence of the fitness values of objective function will be convergent since the stochastic behavior governing the λ-scheme can be explained by a Markov decision process with embedded 3-state (i.e., repeating once, repeating twice, no repeating). Figure 3.7.5 gives the flow chart to express the operations process of $\lambda$ scheme.

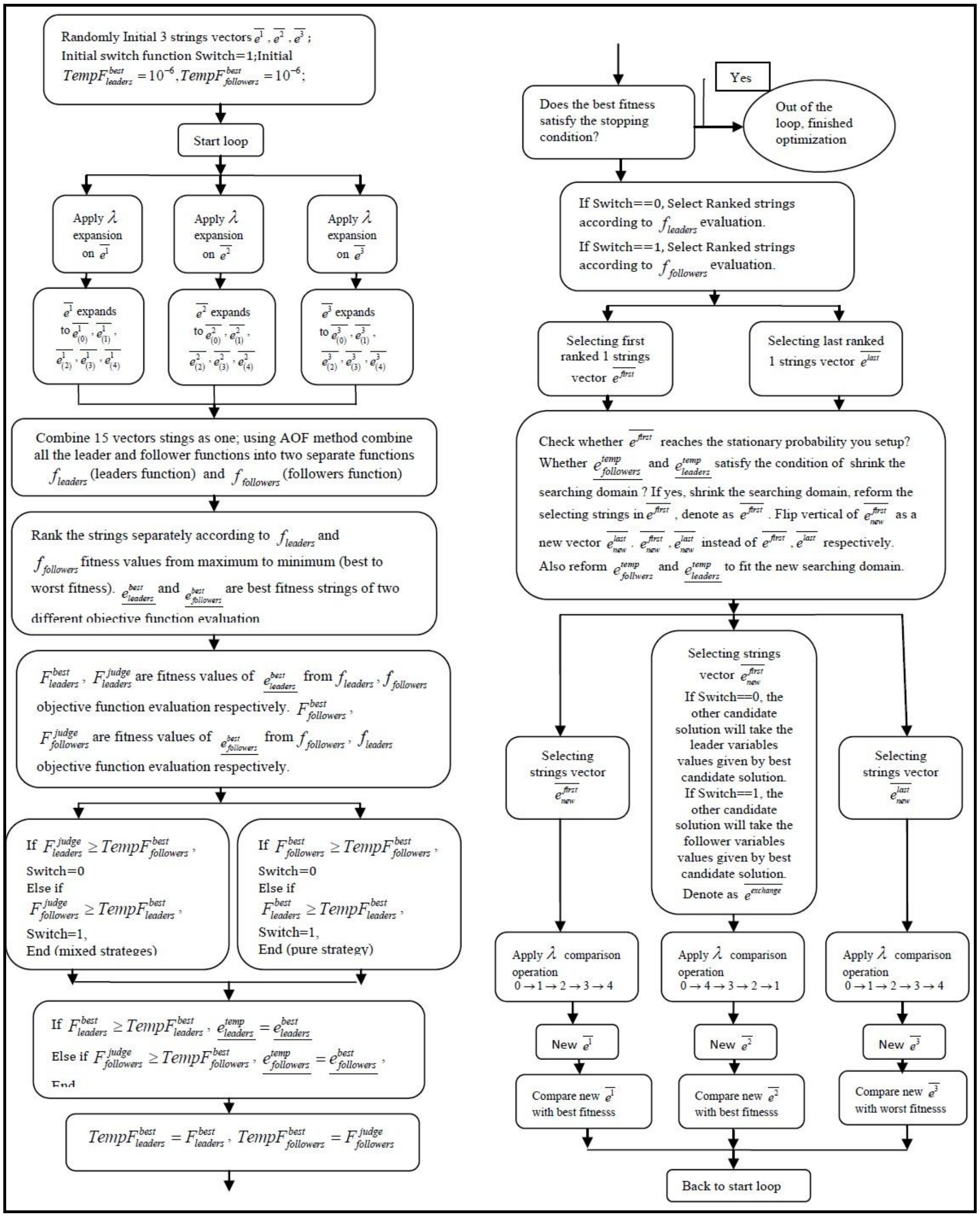


Figure 3.7.5 Flow chart to express the operations process of λ-scheme

As an example, Figure 3.7.6 shows that the λ-scheme GUI software minimizes Rosenbrock function use (5, 4, 2) configuration string vector to represent each variable, the searching domain of two variables is $\left[-10^6, 10^6\right]^2$ . The λ-global optimization scheme spends 10.3436 seconds to achieve the optimal value $f(x_1, x_2) = 4.566\text{E-}10$ with optimal solution $(x_1, x_2) = (1.00000, 0.99999)$.

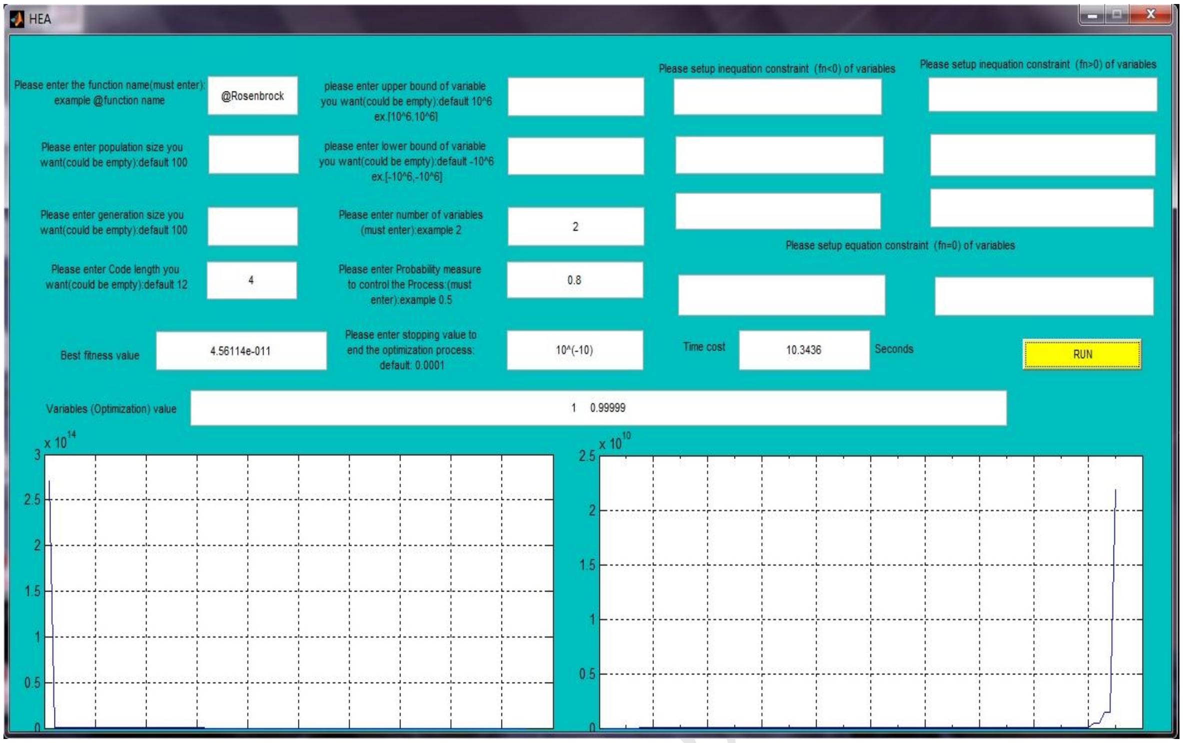



Figure 3.7.6 λ-global optimization scheme GUI software



## 3.7.7 Conclusion

In this paper, after reviewing SML theory and DEAR models, then according to the spirit of the SML algorithm, we focus Bayesian Gaussian process DEAR models as model selection component and also λ-global optimization scheme as computation component for contributing to the SML algorithm;

However, the focus of the paper is explaining the mechanism of DEAR model and λ-scheme, not the applications, although we have applied grey differential equations and DEAR models in reliability research and obtained quick satisfactory results, for details, see reference papers (Guo, 2005; Guo and Love, 2005; Guo, 2007).

However, we must point out the DEAR model fitting process is computationally intensive and delicate. The global optimization searching scheme, say GA, λ-scheme, or others, are a part of the model fitting. Our searching experiences show that the candidate optimal solution is often more than one set. To guarantee the statistically significant parameter estimator, it is necessary to select a solution with standard deviation of the estimator adequately small, such that $0 \notin \left[\hat{\alpha}_i - 3\hat{\sigma}_{\hat{\alpha}_i}, \hat{\alpha}_i + 3\hat{\sigma}_{\hat{\alpha}_i}\right]$.

Our future research on the DEAR model needs to explore the $\sigma_n^2\delta(x,x')$ component, by which the approximation errors caused from difference or accumulated sum replacement of derivative or integral into model variance-covariance construction.

Also, Bundy's incidence calculus (1985, 1986) may guide us to develop a new generation of the SML algorithm, String SML algorithm, which utilizes string vector to represent the family of DEAR models, the members of Gaussian processes, the approximation error bounds and λ-operators to compare and select optimal model(s) according to available data.

# Chapter 4. Discussion

## 4.1 Summary and Assessment

As part of the requirements of a PhD thesis, all of the research works must be fresh and original contributions to the statistical literature. This thesis reports a new statistical machine learning algorithm, named as DEAR-λ-algorithm, in which statistical model selections component is played by DEAR family and the optimization scheme component is the λ-global optimization scheme. Let us summarize them paper by paper.

1. **A naïve five-element string algorithm:**
   In the paper, "A naïve five-element string algorithm", the five-element string idea was formed first time. We admit a fact that five elements, "water, fire, metal, earth, and wood" have deep-rooted Chinese cultural link, nevertheless, the paper is only inspired by the double optimization characteristic in the five-element doctrine, and accordingly designed the operation rules for the five-element string algorithm. In the paper a weighted matrix is suggested, by which a five-element string is linked to an $n$-dimensional vector in Euclidean space. Furthermore, a dual optimal searching scheme is designed into the new algorithm. In certain sense, the five-element string algorithm is an extension from GA's membership set $\{0,1\}$ to the new algorithm's membership set $\{0,1,2,3,4\}$. The string element change enables us getting rid of all of the shortcomings appeared in GA such that mimicking of natural evolution in GA which is hard to control can be avoided, although the stochastic mechanism that governed the five-element algorithm is unknown.
2. **The lambda algorithm:**
   In the paper, "the lambda algorithm", lambda algorithm first time named after the naïve five-element string algorithm in publications. The shrinking search domain method is first time given in the new algorithm. The string specification and the string operations are refined in the new algorithm.
   The second shortcoming from GA is the strings length is too long for the optimization, how could only 3 or 4 bits are used to represent each variable in GA becomes a big question. To address the question the shrinking searching domain method is proposed, which use a certain

stationary probability to control the optimization. Once any column bit of the population reach the stationary, the whole column's bits will be kicked out, at the same time shrink the searching domain, and add another random bits column at each unit's bottom (the best fitness string add 0 at the bottom, to keep the fitness value unchanged). After applying the new method in the lambda algorithm, it becomes feasible to use only 3 or 4 bits to represent each variable. The running speed is accelerated and the optimization result is also much more precise than before.

Dual model in the lambda algorithm is also a big challenge of the old school. The inspiration method is how human thinking about an optimization. Different with today's most popular optimization algorithm, the lambda algorithm not only put its attention on how to make some candidate solutions improve its fitness value, but also considers the usage of worst candidate solution. If thinking about what if the worst thing happened then it would be wise to try to identify the elements causing to the worst situation. This method could help the population of the strings quickly jumping forward into convergence.

3. **Nash –lambda algorithm with applications in safety and reliability:**

In the paper, "Nash–lambda algorithm with applications in safety and reliability", the lambda algorithm is merged with Nash equilibrium solution concept for developing a new algorithm, Nash-lambda algorithm to deal with bi-level non-cooperative game programming optimization. Nash-lambda algorithm allowed program at each loop of optimization evaluate two strategy objective functions. The switches function to decide the rank of all the candidate solutions. If switch=0, then the algorithm according to leader objective function to rank the candidate solutions. If switch=1, then the algorithm according to follower objective function to rank the candidate solutions. Let $TempF_{leaders}^{best}$ and $TempF_{followers}^{best}$ be two variables for recording the best optimization result of leader, and follower objective function in the elapsed optimization. Let $S_{leaders}^{best}$ be the best fitness string of leader objective function at current loop and $S_{followers}^{best}$ be the best follower in the best fitness string of follower objective function at current loop. Denote $F_{leaders}^{best}$, and $F_{leaders}^{judge}$ the fitness values of $S_{leaders}^{best}$ from leader, follower objective function evaluation respectively. Similarly, denote $F_{followers}^{best}$ and $F_{followers}^{judge}$ the fitness values of $S_{leaders}^{best}$ from follower, leader objective function evaluation respectively.

In pure strategy optimization:

$$\begin{aligned}&\text{IF } F_{followers}^{best} \geq TempF_{followers}^{best},\ Switch = 0\\&\text{Else if } F_{leaders}^{best} \geq TempF_{leaders}^{best},\ Switch = 1\\&\text{End}\end{aligned} \tag{1}$$

The above program code means that for leader objective function and follower objective function, each different strategy optimization only allowed jumping once at the algorithm. Once one objective function has a better fitness value then the algorithm must turn to another objective function for a new optimization. If the algorithm runs towards to leader objective function optimization, the selected strings vector $\overline{S_{new}^{first}}$ must let all the candidate solution take the leader variables values given by $S_{leaders}^{best}$. The explanation is that except $S_{leaders}^{best}$, other strings must copy the digits which represent the leader objective function variable $S_{leaders}^{best}$ has. Similarly, if the algorithm runs towards to follower objective function optimization, the selected strings vector $\overline{S_{new}^{first}}$ must let all the candidate solution take the follower variables values given by $S_{follows}^{best}$. The optimization result is after alternative searching "step by step", if one way of the optimization is stopped, which means that this strategy optimization is successful and the pure strategy reaches the Nash equilibrium.

In mixed strategies optimization:

$$\begin{aligned}&\text{IF } F_{followers}^{judge} \geq TempF_{followers}^{best},\ Switch = 0\\&\text{Else if } F_{leaders}^{judge} \geq TempF_{leaders}^{best},\ Switch = 1\\&\text{End}\end{aligned} \tag{2}$$

The above program code means that instead of "step by step" altering optimization, the algorithm allowed optimization jumping at one direction only when the current best fitness is the best fitness of leader and follower objective function, if the solution satisfies with above condition, then the algorithm allowed the optimization towards to another way. The optimization result is more balanced in this way, which can give many more Nash equilibriums for different strategies.

**4. Lambda algorithm and maximum likelihood estimation:**

In the paper, "lambda algorithm and maximum likelihood estimation", it is first time to merge the lambda global optimization searching to the popular statistical estimation procedure. Maximum likelihood estimation is traditionally derivative-oriented, while the lambda algorithm is derivative-free scheme. Standard maximum likelihood search is using the Newton-Raphson procedure, see Lawless (1982). Notice that a local optimal solution may be resulted in although statistically it is a maximum likelihood estimator. But this "local" solution may not give a meaningful solution in practical circumstances. Another issue is in the derivative-oriented approach, the calculations of the first-order and the second partial derivatives are heavy and tedious, which may cause overflow and underflow problems. The new merged algorithm avoids the two fundamental problems. During the searching global solution, a series of local optimum can be recorded and thus an appropriate (in practical sense) optimum may be selected as the final maximum likelihood estimator. Although in merged algorithm, the first-order and the second partial derivatives are derived but at calculated them the end of searching for final checking if the optimal criterion is met.

Besides, both of structures and operations of lambda algorithm have a big change after the "the lambda algorithm" paper.

The dual operator $\lambda$ and $\lambda^{-1}$ instead of $\lambda$ operator execute the $\lambda$ comparison operation in the population strings.

In the merged algorithm, the $\lambda$ comparison operation with population is also changed. After the population of strings sort by their fitness values (from minimum to maximum), we used to compare each string with other four strings ranked below of it. The purpose is to separate unrepeated, once repeated, twice repeated, third time repeated and fourth time repeated information in different state, to join with new bits. But lots of times, before the column bits going to convergence, there are only 2 different bits could be involved with stationary problem in one column, then unrepeated, once repeated and twice repeated information is enough to reach different situation we met, because the bit set is just contained of 5 different elements.

5. **Decision theory under general uncertainty:**

In this paper, the basic elements of decision analysis under general uncertainty are defined and discussed. The motivation to develop the general uncertain theory was inspired by such observations that the real world surrounding us is not as simple as people often think. Uncertainty is intrinsic and diversified in form. For example, vagueness is a different form of uncertainty from randomness, and enters more and more into today's industrial environments, as Carvalho and Machado (2006) have commented, "In a global market, companies must deal with a high rate of changes in business environment… The parameters, variables and restrictions of the production system are inherently vagueness."

The existing statistical decision theory is a framework with a probabilistic foundation, which admits random uncertainty about the real world and human thinking. It may not guide people toward corrected decision when a different form of uncertainty appears. Random uncertainty is merely a special form of uncertainty and hence using probabilistic statistical decision theory to deal with it may be oversimplified. Therefore, in this paper, an axiomatic uncertain measure theoretical framework is introduced and the essential mechanism in formulating a general uncertainty decision theory is explored.

6. **Bayesian uncertainty decision theory:**

In this paper, Bayesian uncertain decision analysis is proposed. If examining the uncertainty theory and probabilistic Bayesian theory, some commonality between them can be identified: (1) Bayesian sometimes be considered as subjective statistics, uncertainty theory use expert's knowledge as data information, also very subjective. Both emphasize expert's knowledge-prior evidence. (2) Bayesian use sampled data, uncertainty theory use expert's knowledge, both of them emphasize evidences. (3) Both emphasize the merging two sources of evidences for generating post-knowledge. (4) Both emphasize the impreciseness in data or knowledge. (5) Both emphasize the representations in distributional form-the posterior density or posterior uncertainty distribution.

However there are fundamental differences between them: (1) Probabilistic Bayesian combines prior and likelihood via the Bayes formula, the integration involvements. (2) Probabilistic Bayesian utilizes likelihood (*i.i.d.*) for handling sampled data information. (3) Uncertainty Bayesian utilize the Liu's product measure axiom (2007, 2010, 2011) for combining prior knowledge and expert's experiment knowledge. (4) Uncertainty Bayesian may utilize copula-linked marginal's uncertainty multivariate distribution for handling expert's knowledge.

Table 4.1.1 Bayes formula under probability theory and uncertainty theory

| | Probability theory | Uncertainty theory |
|---|---|---|
| Bayes formula | $g\left(\theta \mid \underline{x}\right)=\dfrac{f\left(\underline{x} \mid \theta\right)g\left(\theta\right)}{\int_{\theta} f\left(\underline{x} \mid \theta\right)g\left(\theta\right)d\theta}$ | $\Psi\left(\theta \mid \underline{x}\right)=\dfrac{\Psi\left(\underline{x},\theta\right)}{\Psi\left(\underline{x}\right)}=\dfrac{\Psi\left(\underline{x},\theta\right)}{\sup_{\theta}\left(\Psi\left(\underline{x},\theta\right)\right)}$ |
| Parameter | $\theta \in \Theta$ | $\theta \in \Theta$ |
| Observations | $F\left(x_1\right)\leq F\left(x_2\right),\ \text{for } \forall x_1 < x_2$ | $x$ |

According to Table 4.1.1, it is obvious that in traditional probability the integration of density function to calculate the marginal distribution. In uncertainty Bayesian formula, instead of use integration of density, we use uncertainty product measure axiom to calculate the marginal distribution in Bayesian formula. In the paper, an illustrative example is given in detail.

7. **Probabilistic DEAR modeling:**

In this paper a new DEAR-λ-algorithm is developed as contributions to the statistical machine learning algorithm.

Differential equation associated regression (abbreviated as DEAR) is a flexible and powerful data mining modeling approach, which is intended to catch up the first-order non-linear trend (i.e., regularity) governing the behavior of the data under investigation. DEAR modeling is a formal mathematical-statistical representation of the so-called grey differential equation model. It should be pointed out that DEAR models were originally proposed on the random fuzzy theoretical foundation. Nevertheless, DEAR models can be defined on any measure theoretic platform, for example, probabilistic, fuzzy, or uncertain measure foundation as long as the model and approximation two constituting components are appropriately specified.

In this paper, we re-examine the compositional elements of DEAR models and the potential model selection portfolio in the statistical machine learning (SML) algorithm developments. Then the differential equation backed DEAR may contribute to the statistical machine learning algorithm significantly, particularly, in developing robot movement system, where the motion laws are expressed directly by a set of differential equations. Under a statistical decision theoretical framework, a DEAR model which is constituted by a random function with a linear difference equation-wise regression as the central tendency and a variance bound specified by Gaussian error analysis theory is developed delicately, in which the prior distribution will be facilitated by a Gaussian process such that the replication of sampling for estimating the weight matrix will be avoided. Therefore the model selection compositional elements of the statistical machine learning algorithm is addressed but also the optimization scheme, which is called λ-global optimization scheme is addressed too.

In this paper, based on statistical decision theory, a new probabilistic DEAR model is established, which is effectively utilizes the members from a Gaussian process as the error terms (Rasmussen and Williams, 2006; Snelson et al., 2004; Williams and Rasmussen, 1996). Hence, the autocovariance matrix can be facilitated part of the DEAR error structure. That is, the Gaussian process DEAR model will merge data-design matrix and the intrinsic autocovariance matrix together to address the regression coefficient estimation problem.

The new DEAR regression model will be accessible to any data mining algorithms and is suitable to computer system self learning and adjusting because DEAR algorithm becomes a weighted regression, while the weight matrix is just the intrinsic autocovariance matrix plus the Gaussian approximation error matrix. The dimension of the weight matrix depends upon data size, and the coefficient estimator depends on the data-design matrix $X$ and weight matrix $\Sigma$, i.e., $X\Sigma^{-1}X$.

## 4.2 Further discussion on the stochastic behaviour of λ-scheme

It is fair to say that the developments in λ-global optimization scheme have achieved its target. However, the stochastic mechanism governing the λ-searching process is still remaining in an un-addressed issue.

To further understand the underlying mechanism of λ-scheme, let us analyze the 3-states Markov decision model for seeking the mathematical explanation.



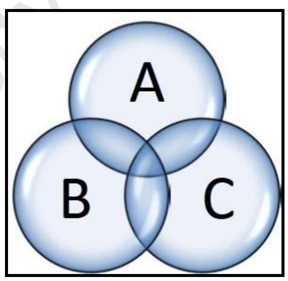



Figure 4.2.1 Sets *A*, *B* and *C* Venn diagram

Let *A*, *B* and *C* be 3 sets, a Venn diagram shows the intersection pattern. (See Figure 4.2.1). Let set *C* represent the objective set under investigation. Set $C$ can be partition into mutually disjoint subsets of three types, Type One: no repeating $C-A\cap C-B\cap C+A\cap B\cap C$, Type Two, repeat once, subset $(B\cap C)-(A\cap B\cap C)$, and subset $(A\cap C)-(A\cap B\cap C)$, and Type Three, repeat twice, $A\cap B\cap C$.

We regard the separation of the 3 different exclusive types of subsets from set *C* as a three-state Markov decision process. (See Figure 4.2.2)

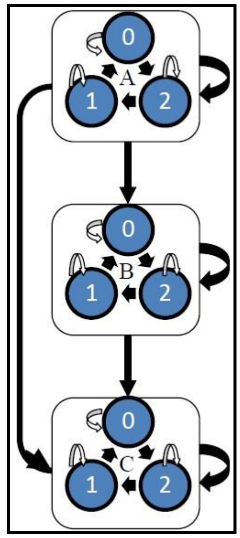



Figure 4.2.2 Three-state Markov decision process



In the three-state Markov decision model, the states are *A*, *B* and *C*. Within each state, another 3-state Markov chain is embedded, in which 0, 2 and 1 constitute of the embedded Markov chain states, the state change follows a clockwise manner. Initial set *A*, *B* and *C* information all store at their own sub-state 0 respectively, another 2 sub-state 1 and 2 are empty.

1. Compare state *A* sub-state 0 with state *B* sub-state 0. In state *B*, if sub-state 0 contains information about $A \cap B$, then move subset $A \cap B$ into sub-state 2, keep subset $B-(A \cap B)$ in sub-state 0.
2. Compare state *A* sub-state 0 with state *C* sub-state 0. In state *C*, if sub-state 0 contains information about $A \cap C$, then move subset $A \cap C$ into sub-state 2, keep subset $C-(A \cap C)$ in sub-state 0.
3. Compare state *B* sub-state 0 and 2 with state *C* sub-state 0 and 2. In state *C* sub-state 0, if sub-state 0 contains information about $(B \cap C)-(A \cap B \cap C)$, then move subset $(B \cap C)-(A \cap B \cap C)$ into subset 2, keep $C-A \cap C-B \cap C+A \cap B \cap C$ in state 0. Sub-state 2 becomes $(A \cap C)+(B \cap C)-(A \cap B \cap C)$. At the same time in state *C* sub-state 2, if sub-state 2 contains information about $A \cap B \cap C$, then move subset $A \cap B \cap C$ into sub-state 1, keep $(A \cap C)+(B \cap C)-(A \cap B \cap C)-(A \cap B \cap C)$ in sub-state 2.

In state C, we have 2 times repeated information $A \cap B \cap C$ in sub-state 1; have only 1 time repeated information $((A \cap C)-(A \cap B \cap C))+((B \cap C)-(A \cap B \cap C))$ in sub-state 2; have no repeated information $(C-A \cap C-B \cap C)+(A \cap B \cap C)$ in sub-state 0. Now twice repeated information subset, once time repeated information subset and no repeated information subset embedded in set $C$ are completely separated.

In order to make this 3-state Markov decision model easier to understand, we give an example of 3-element string (for which element set is $\Theta = \{0,1,2\}$). $A$, $B$ and $C$ 3 strings are list as following. In string $C$, twice repeated elements are recorded with blue color number, once repeated elements are recorded with red color number, no repeated elements are recorded with green color number.

***A***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 0 | 1 | 2 | 1 | 1 | 2 | 0 | 2 | 1 | 2 |

***B***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 0 | 1 | 0 | 2 | 2 | 1 | 1 | 1 | 0 | 1 |

***C***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 0 | 1 | 2 | 1 | 2 | 1 | 2 | 0 | 2 | 0 |

The value of the element $e_j$ follows $0 \to 1 \to 2 \to 0$ criterion to change, which is defined as $\lambda(e_j)$:

```
For j=1:1:12
If e^0_{A(j)} ≡ e^0_{B(j)}
e^0_{B(j)} ← λ⌊e^0_{A(j)}⌋
ELSE IF e^0_{C(j)} ≡ e^0_{A(j)}
e^0_{C(j)} ← λ⌊e^0_{A(j)}⌋
ELSE IF e^0_{C(j)} ≡ e^0_{B(j)}
e^0_{C(j)} ← λ⌊e^0_{B(j)}⌋
END
```

After executing the above program, 3 strings $A$, $B$ and $C$ will become as follows:

***A***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 0 | 1 | 2 | 1 | 1 | 2 | 0 | 2 | 1 | 2 |

***B***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 2 | 1 | 0 | 0 | 2 | 1 | 1 | 0 | 1 |

***C***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 2 | 0 | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 0 |

Expansion string $C$ in other two states 1 and 2 we will have

***C-0***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 2 | 0 | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 0 |

***C-1***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

***C-2***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 2 | 1 | 2 |

As the 3-state Markov decision model showed in Figure 4.2.2, in above result, we cluster twice repeated cells (blue colored) in sub-state *C-1*; cluster once repeated cells (red colored) in sub-state *C-2*; keep no repeated cells (green colored) in sub-state *C-0*.

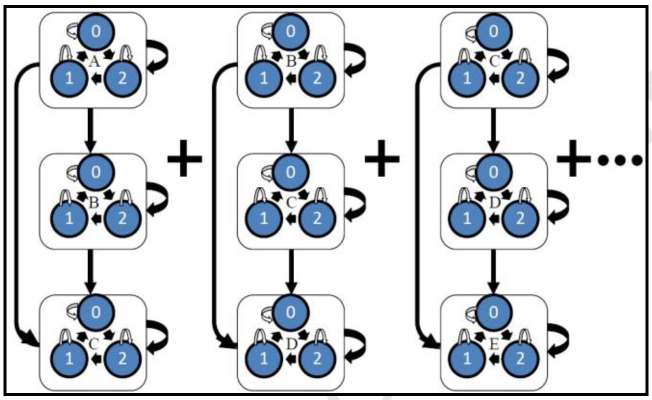



Figure 4.2.3 The overlap of many 3-state Markov decision models

In order to make many strings overlapped after original string *C* also achieving the effect of separate twice repeated cells, once repeated cells and no repeated cells in different sub-states, we write a program for achieving an effect equal to summing many of 3-state Markov decision models together. (See figure 4.2.3)

In the λ-scheme, we use five-element set $\Theta = \{0,1,2,3,4\}$ to construct a string, so the number of the substates are not 3 but 5 (i.e., substate 0 to 4). The overlap of many 3-state Markov decision models, by using 5-element (sub-state) system is listed as below (see Figure 4.2.4). In Figure 4.2.4, we have 3 columns of the process, where each column of the process has its state changing effect is determined by summing of many 3-state Markov decision models. Different columns are standard as different loops in the λ-scheme program.

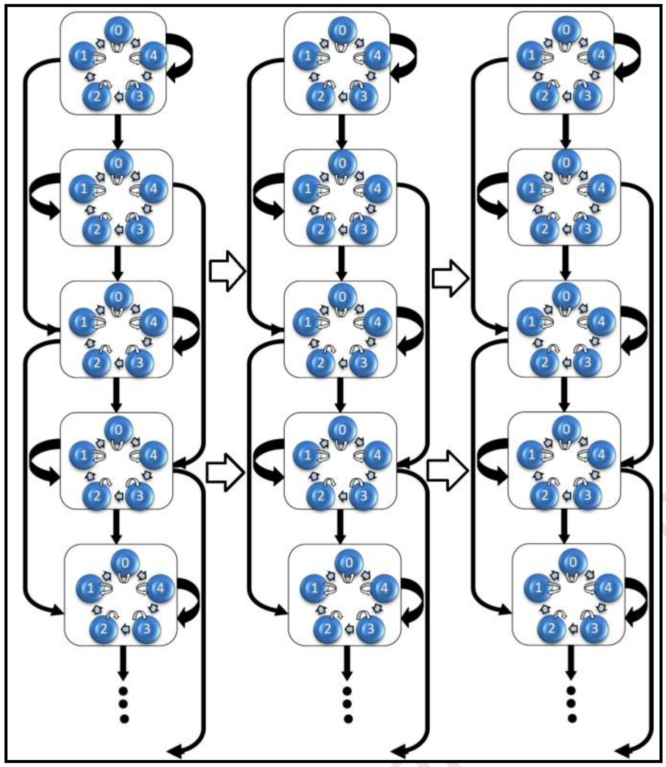


Figure 4.2.4 λ comparison and expansion operation in λ-scheme

Each column process in Figure 4.2.4 could be simply understood as the combination of λ comparison and expansion operations. λ comparison operation of the first kind means that the value of the element $e_i$ follows $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ criterion to change. Let $\underline{e}_1^1$, $\underline{e}_2^1$, ⋯, $\underline{e}_{100}^1$ are strings in string vector $\overline{e^1}$. Let $e_{i(1)}^1, e_{i(2)}^1, \cdots, e_{i(12)}^1$ are elements in any string $\underline{e}_{1,i}$ respectively, $i = 1, 2, \cdots, 100$. Then

For i=1:1:100-2
For j=1:1:12
If $e_{i(j)}^1 \equiv e_{(i+1)(j)}^1$
$e_{(i+1)(j)}^1 \leftarrow \lambda\left[e_{i(j)}^1\right]$
ELSE
IF $e_{i(j)}^1 \equiv e_{(i+2)j}^1$
$e_{(i+2)(j)}^1 \leftarrow \lambda\left[e_{i(j)}^1\right]$
END

The above program $e^1_{(i+1)(j)} \leftarrow \lambda\left[e^1_{i(j)}\right]$ and $e^1_{(i+2)(j)} \leftarrow \lambda\left[e^1_{i(j)}\right]$ procedure, is not meaning immediately evaluate $\lambda\left[e^1_{i(j)}\right]$ value and give it to $e^1_{(i+1)(j)}$, $e^1_{(i+2)(j)}$ respectively. Actually, the λ-scheme has a recording matrix, which is used to record how many times each cell $e^1_{i(j)}$ need to execute $\lambda[\cdot]$ operation, the original string vector $\overline{e^1}$ will not change any value before the end of the whole comparison operations completed. About the record matrix, we give an example as below:

***A***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 0 | 1 | 2 | 1 | 1 | 2 | 0 | 2 | 1 | 2 |

***B***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 0 | 1 | 0 | 2 | 2 | 1 | 1 | 1 | 0 | 1 |

***C***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 0 | 1 | 2 | 1 | 2 | 1 | 2 | 0 | 2 | 0 |

The recording matrix of string *C* will write as follows:

Recording matrix of ***C*** is

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

Each cell of recording matrix records how many times each cell of string vector executes $\lambda[\cdot]$ operation.

By the end of whole comparison, string vector $\overline{e^1}$ will according record matrix to change its values. λ expansion operation will accord string vector $\overline{e^1}$ (sub-state 0) to generate other 4 sub-state values. Then we will see the no repeated cells keep stay at sub-state 0, once repeated cells move to sub-state 4, twice repeated cells move to sub-state 3. The system has same effect as summing many 3-state Markov decision models.

## 4.3 Future Developments

In this thesis, a new statistical machine learning algorithm – DEAR- λ-algorithm is developed. The developments can be divided into aspects: λ-global optimization scheme, probabilistic DEAR models, and uncertainty Bayesian decision theory.

About the λ-global optimization scheme, the stochastic mechanism governing the searching process requires more efforts to dig out. The probability network, which involves conditional independence with Markov property framework corresponds to d-separation in graphoid framework, will be investigated.

About probabilistic DEAR modeling family, it is necessary to enlarge and grouping the models such that the model selections will be more efficient and flexible.

About the uncertainty decision theory, the Bayesian uncertain decision method only could join with small data set with the prior-expert's knowledge. In the future, we need to consider of how to involve large data sample with Bayesian uncertain formula into analysis. Facing the problem from various environments, uncertainty decision theory has a long way to go.

# Chapter 5. Conclusion

The overall aim of the research is to create a new statistical machine learning algorithm, named as Differential Equations Associated Regression Modelling (DEAR)-λ-algorithm. To achieve the overall objective of this thesis, seven journal papers are listed in Chapter 3, two of them are EI indexed. The thesis papers were classified into 3 categories according to their aims:

1. First aim is to use statistical and mathematical methods to design a new global optimization scheme (i.e., a global optimization algorithm), λ-algorithm. In the paper, "A naïve five-element string algorithm", the naïve five-element string algorithm is the first time we published our research results of λ-algorithm as a published work. After a long time of methods developing, the five-element string algorithm becomes today's λ-algorithm. In the paper, "the lambda algorithm", lambda algorithm first time named after the naïve five-element string algorithm in publications. The basic algorithm structure and mathematical explanation of the algorithm are illustrated in this paper. In the sequential paper, "Nash–lambda algorithm with applications in safety and reliability", the λ-algorithm is merged with Nash equilibrium solution for dealing with bi-level non-cooperative game programming optimization. In the paper, "lambda algorithm and maximum likelihood estimation", the paper work is first time using the λ-lambda optimization for refining certain statistical estimation methods. Compare with traditional computation method, maximum likelihood estimation with λ search will be more quickly and easier, especially when we met multiple parameters estimation or very hard computation problems. In section 4.2 Further discussion on the stochastic behaviour of λ-scheme, a 3-states Markov decision model as the mathematical explanation of automatic classification of different information in strings. Through testing of classified information, the strings towards to updating their fitness values better and better to finally achieve the optimization.
2. The second aim is to build up uncertainty decision theory and modelling to deal with general uncertainty data. In the papers, "decision theory under general uncertainty", we point out the uncertainty of the real world is too diversified which the random uncertainty may be oversimplified to measure it. In this paper we argue the necessity to examine current decision foundation under the "Made in Japan" crisis, introduce an axiomatic uncertain measure

theoretical framework and review framework, the measure theoretical statistical decision theory for preparing the new general uncertainty decision theory developments. In the Paper, we further discuss the basic elements and its intrinsic feature of the uncertainty decision theory by comparing to the statistical decision theory. Particularly, the fundamental role and the essential form of uncertainty distributions are revealed. Accordingly, the general uncertainty decision - making approach in discrete and continuous uncertainty environments including Maximum Uncertainty Principle (MUP)  Bayesian approach, are developed respectively. In the paper, "Bayesian uncertainty decision theory", we apply Bayesian decision method under uncertainty measure theory and created the Bayesian uncertainty decision theory.

3. The third aim is to use the new knowledge in developing uncertain canonical process regression to develop a probabilistic differential equations associate regression, which is a continue research of my master thesis "statistical-grey consistency of grey differential equation models". More importantly, the new modelling family simplifies the DEAR model and make it being suitable to data mining task. At this point, DEAR modelling family can facilitate the statistical modelling selections such that the new statistical machine learning algorithm is in shape. For details, see a series of papers: Dai et al. (2011), R. Guo et al. (2011), and D. Guo et al. (2011).

Future research will continue focusing on the improvements of the new statistical machine learning algorithm, DEAR λ-algorithm because statistical machine learning is the merging frontier of modern statistical science and computer science.

# Appendix A: Responses and Corrections

| **Professor Dr K Kolowrocki Comments:** |
|---|
| 1. Concerning Section 3.1: The deficiency of this section on giving Lemma (5) and Propositions (13, 15, and 16) without profs and references should be clarified by the candidate to the COA. |
| **Contents in Examined Thesis:** |
| **Lemma 5:** Let the system state be $\underline{x} \in \mathbb{D} \subset \mathbb{R}^n$, and $\underline{e}$ be a string representation (of the system state) with element set size $s$ and string length $n(s+1)$. Let $\underline{u}_{\min} \leq \underline{x} \in \mathbb{D}$, $\underline{u}_{\max} \geq \underline{x} \in \mathbb{D}$, and $u_r = \max\limits_{1 \leq i \leq n} \{u_{\max,i} - u_{\min,i}\}$. The weight matrix $O = (o_{ij})_{n \times nu} = (\underline{o}_1^T, \underline{o}_2^T, \cdots, \underline{o}_n^T)^T$ with the $i^{th}$ row vector $\underline{o}_i^T$ having a form $\left(0,0,\cdots,0,\cdots,\frac{S^s}{S^{s+1}},\frac{S^{s-1}}{S^{s+1}},\cdots,\frac{S^0}{S^{s+1}},\cdots,0,0,\cdots,0\right)$ where the nonzero weights are located at the $i^{th}$ segment. Then the system state is a linear transformation of the $s$-element string representation $\underline{x} = \underline{u}_{\min} + u_r O \underline{e}$ |
| **Response and Corrections:** |
| **Proof Lemma5:**<br>Select any $\underline{e}_i \subset \underline{e}$,<br>According to presentation of an unit of string<br>$\underline{e}_i \in [0,0,\cdots,0 \sim S-1, S-1,\cdots,S-1]$<br>Let $\underline{x}'_i = \underline{o}_i^T \cdot \underline{e}_i$<br>$\therefore \underline{x}'_i \in \left[0,0,\cdots,0 \sim \frac{S^s \cdot (S-1)}{S^{s+1}}, \frac{S^{s-1} \cdot (S-1)}{S^{s+1}}, \cdots \frac{S^0 \cdot (S-1)}{S^{s+1}}\right]$ $\underline{x}'_i \in \left[0, \frac{S^s \cdot (S-1) + S^{s-1} \cdot (S-1) + \cdots + S^0 \cdot (S-1)}{S^{s+1}}\right]$<br>$\underline{x}'_i \in \left[0, \frac{S^{s+1}-1}{S^{s+1}}\right]$<br>$\underline{x}'_i \in \left[0, 1-\frac{1}{S^{s+1}}\right]$<br>hence that $u_r \cdot \underline{x}'_i \in \left[0, (u_{\max,i} - u_{\min,i}) \cdot (1-\frac{1}{S^{s+1}})\right]$ $\therefore u_r \underline{o}_i^T \underline{e}_i \in \left[0, (u_{\max,i} - u_{\min,i}) \cdot (1-\frac{1}{S^{s+1}})\right]$<br>Let $\underline{x}''_i = u_{\min,i} + u_r \cdot \underline{x}'_i$<br>$\therefore \underline{x}''_i \in \left[u_{\min,i}, (u_{\max,i} - u_{\min,i}) \cdot (1-\frac{1}{S^{s+1}}) + u_{\min,i}\right]$ Because $\underline{x}_i \in [u_{\min,i}, u_{\max,i}]$ |

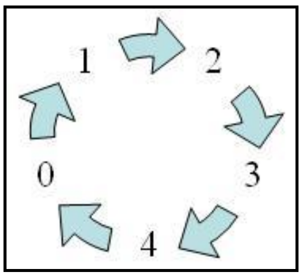
| |
|---|
| when $s \to +\infty$, $\frac{1}{S^{s+1}} \to 0$<br>$\underline{x}"_i \approx \underline{x}_i$<br>In the algorithm, the system could make $s \to +\infty$, if the final solution doesn't satisfied our request and going to convergence,<br>So, $\underline{x} = \underline{u}_{\min} + u_r O\underline{e}$ is almost sure. |
| **Contents in Examined Thesis:** |
| **Proposition 13:** For any given five-element string $\underline{e}$, the five-time $\lambda[\ ]$ operated strings form a string cycle. In other words, $\left\{\underline{e}, \lambda^{(1)}[\underline{e}], \lambda^{(2)}[\underline{e}], \lambda^{(3)}[\underline{e}], \lambda^{(4)}[\underline{e}]\right\}$ is a string cycle. |
| **Response and Corrections:** |
| **Proof Proposition 13:**<br>Select any element from a string $e_i \in \underline{e}$,<br>$\lambda[e] = \begin{cases} e+1 \text{ if } e \in \{0,1,2,3,4\} \\ 0 \quad\quad \text{if } e = 4 \end{cases}$<br>Let $e_i = 0$ $\left\{e_i, \lambda^{(1)}[e_i], \lambda^{(2)}[e_i], \lambda^{(3)}[e_i], \lambda^{(4)}[e_i]\right\} = \{0,1,2,3,4\}$<br>Let $e_i = 1$ $\left\{e_i, \lambda^{(1)}[e_i], \lambda^{(2)}[e_i], \lambda^{(3)}[e_i], \lambda^{(4)}[e_i]\right\} = \{1,2,3,4,0\}$<br>Let $e_i = 2$ $\left\{e_i, \lambda^{(1)}[e_i], \lambda^{(2)}[e_i], \lambda^{(3)}[e_i], \lambda^{(4)}[e_i]\right\} = \{2,3,4,0,1\}$<br>Let $e_i = 3$ $\left\{e_i, \lambda^{(1)}[e_i], \lambda^{(2)}[e_i], \lambda^{(3)}[e_i], \lambda^{(4)}[e_i]\right\} = \{3,4,0,1,2\}$<br>Let $e_i = 4$ $\left\{e_i, \lambda^{(1)}[e_i], \lambda^{(2)}[e_i], \lambda^{(3)}[e_i], \lambda^{(4)}[e_i]\right\} = \{4,0,1,2,3\}$<br>$\therefore$ for any $e_i$, $\left\{e_i, \lambda^{(1)}[e_i], \lambda^{(2)}[e_i], \lambda^{(3)}[e_i], \lambda^{(4)}[e_i]\right\}$ is a string cycle.<br>Hence that<br>For a string, $\underline{e}$, $\left\{\underline{e}, \lambda^{(1)}[\underline{e}], \lambda^{(2)}[\underline{e}], \lambda^{(3)}[\underline{e}], \lambda^{(4)}[\underline{e}]\right\}$ is a string cycle. |
| **Contents in Examined Thesis:** |
| **Proposition 15:** Let<br>$f_{\min} = \min\left\{f(\underline{x}), f(\underline{x}^{(1)}), f(\underline{x}^{(2)}), f(\underline{x}^{(3)}), f(\underline{x}^{(4)})\right\}$<br>$f_{\max} = \max\left\{f(\underline{x}), f(\underline{x}^{(1)}), f(\underline{x}^{(2)}), f(\underline{x}^{(3)}), f(\underline{x}^{(4)})\right\}$ |

| |
|---|
| Then the objective function cycle will demonstrate three patterns: (i) $f(\underline{x}) = f_{\min}$, i.e., the remaining four objective function values are above the cycle starting value $f(\underline{x})$; (ii) $f(\underline{x}) = f_{\max}$, i.e., the cycle starting value $f(\underline{x})$; (iii) $f_{\min} \le f(\underline{x}) \le f_{\max}$, i.e., the cycle starting value $f(x)$ falls between cycle minimum and maximum. |
| **Response and Corrections:** |
| **Proof: Proposition 15:**<br>$\because \underline{x} = \underline{u}_{\min} + u_r O \underline{e}$<br>$\therefore \underline{e}$ is a linear mapping of the input variables $\underline{x}$<br>The cycle behavior generates different values of the strings.<br>$\{\underline{e}, \lambda^{(1)}[\underline{e}], \lambda^{(2)}[\underline{e}], \lambda^{(3)}[\underline{e}], \lambda^{(4)}[\underline{e}]\}$<br>which mapping to the real line to get different values of<br>$\{\underline{x}, \lambda^{(1)}[\underline{x}], \lambda^{(2)}[\underline{x}], \lambda^{(3)}[\underline{x}], \lambda^{(4)}[\underline{x}]\}$<br>when $\left\{f(\underline{x}), f\left(\underline{x}^{(1)}\right), f\left(\underline{x}^{(2)}\right), f\left(\underline{x}^{(3)}\right), f\left(\underline{x}^{(4)}\right)\right\}$<br>could generates more than 3 different values, which satisfied the Proposition 15. |
| **Contents in Examined Thesis:** |
| **Proposition 16:** The weight matrix $O$ in string and system state linking equation $\underline{x} = \underline{u}_{\min} + u_r O \underline{e}$ reveals the ever-changing and controllable character of five element string representation. And the three cycle patterns of objective function values with respect to string cycles reveal that $\lambda[\ \ ]$ operations guarantee the chance for global optimum searching. |
| **Response and Corrections:** |
| **Proof: Proposition 16:**<br>$\because \underline{x} = \underline{u}_{\min} + u_r O \underline{e}$<br>If element set size $s \to \infty$<br>$\underline{e}$ will be totally mapping to the real line according to $\underline{x} \in [\underline{u}_{\min}, \underline{u}_{\max}]$<br>So, if strings population size $N \to \infty$, any position of string $e_i$ get at least one time chance repeated with other string in same position.<br>Then $\lambda[\ \ ]$ will guarantee the chance for global optimum searching. |

| **Professor Dr K Kolowrocki Comments:** |
| --- |
| Concerning Section 3.2: The deficiency of this section on giving Theorem (9) and Propositions (17) without profs and references should be clarified by the candidate to the COA. |
| **Contents in Examined Thesis:** |
| **Theorem 9:** If G is strongly connected then there is a unique stationary distribution π for M. Moreover, this distribution satisfies $\pi_j > 0$ for all $j \in N$. |
| **Response and Corrections:** |

**Proof Theorem 9:**

The G graph is show as below:




From G graph we can see, G is an irreducible and also **aperiodic directed graph.**

Definition of aperiodic graph: In the mathematical area of graph theory, a directed graph is said to be aperiodic if there is no integer k > 1 that divides the length of every cycle of the graph. Equivalently, a graph is aperiodic if the greatest common divisor of the lengths of its cycles is one; this greatest common divisor for a graph G is called the period of G.（http://en.wikipedia.org/wiki/Aperiodic_graph）

Because G graph is a directed graph and also aperiodic. Let $P$ be the transition matrix, which satisfied condition $\lim_{t\to\infty} p^t = p^\infty$,

$$p = \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 & \pi_5 \\ \pi_5 & \pi_1 & \pi_2 & \pi_3 & \pi_4 \\ \pi_4 & \pi_5 & \pi_1 & \pi_2 & \pi_3 \\ \pi_3 & \pi_4 & \pi_5 & \pi_1 & \pi_2 \\ \pi_2 & \pi_3 & \pi_4 & \pi_5 & \pi_1 \end{bmatrix}$$

Let $\pi = (\pi_1 \quad \pi_2 \quad \pi_3 \quad \pi_4 \quad \pi_5)$,

Because $p^\infty p = \left(\lim_{t\to\infty} p^t\right) p = \lim_{t\to\infty} p^{t+1} = p^\infty$

Then $\pi$ is a stationary distribution

For any initial distribution $X_0$,

the sequence $X_t = X_{t-1}p = X_0 p^t$

converges to $\pi$,

$$\lim_{t\to\infty} X_t = \lim_{t\to\infty}\left(X_0 p^t\right) = X_0\left(\lim_{t\to\infty} p^t\right) = X_0 p^\infty = \sum_{i=1}^{n} X_0^i \pi = \left(\sum_{i=1}^{n} X_0^i\right)\pi = \pi$$

Thus, if the chain is irreducible and aperiodic, there is a unique limit, stationary distribution.

**Contents in Examined Thesis:**

**Proposition 17:** Let

$$f_{\min} = \min\left\{f(\underline{x}), f\left(\underline{x}^{(1)}\right), f\left(\underline{x}^{(2)}\right), f\left(\underline{x}^{(3)}\right), f\left(\underline{x}^{(4)}\right)\right\}$$
$$f_{\max} = \max\left\{f(\underline{x}), f\left(\underline{x}^{(1)}\right), f\left(\underline{x}^{(2)}\right), f\left(\underline{x}^{(3)}\right), f\left(\underline{x}^{(4)}\right)\right\}$$

Then the objective function cycle will demonstrate three patterns: (i) $f(\underline{x}) = f_{\min}$, i.e., the remaining four objective function values are above the cycle starting value $f(\underline{x})$; (ii) $f(\underline{x}) = f_{\max}$, i.e., the remaining four objective function values are below the cycle starting value $f(\underline{x})$; (iii) $f_{\min} \le f(x) \le f_{\max}$, i.e., the cycle starting value $f(\underline{x})$ falls between cycle minimum and maximum.

**Response and Corrections:**

**Proof: Proposition 17:**

$\because \underline{x} = \underline{u}_{\min} + u_r O \underline{e}$

$\therefore \underline{e}$ is a linear mapping of the input variables $\underline{x}$

The cycle behavior generates different values of the strings.

$\left\{\underline{e}, \lambda^{(1)}[\underline{e}], \lambda^{(2)}[\underline{e}], \lambda^{(3)}[\underline{e}], \lambda^{(4)}[\underline{e}]\right\}$

which mapping to the real line to get different values of

$\left\{\underline{x}, \lambda^{(1)}[\underline{x}], \lambda^{(2)}[\underline{x}], \lambda^{(3)}[\underline{x}], \lambda^{(4)}[\underline{x}]\right\}$

when $\left\{f(\underline{x}), f\left(\underline{x}^{(1)}\right), f\left(\underline{x}^{(2)}\right), f\left(\underline{x}^{(3)}\right), f\left(\underline{x}^{(4)}\right)\right\}$

could generates more than 3 different values, which satisfied the Proposition 17.

**Professor Dr K Kolowrocki Comments:**

Concerning Section 3.3: More detailed describing by the candidate the possibility of application of the Nash-Lambda algorithm to real safety and reliability problem would increase the thesis applicable value.

**Response and Corrections:**

1. The thesis is written in a collection of published papers style, the published journal paper do have a limitation of how many pages of writing. That's why the paper is limited with examples.
2. The Nash-Lambda algorithm is a new method, which we need give everyone a clearly explanation of its dynamic system, then a short example to show the new algorithm is actually works with real application. About more real applications we need to do more research, and write papers in the future.

**Professor Dr K Kolowrocki Comments:**

Concerning Section 3.4 and 3.5: In spite of the papers are typical review ones, without own results in the field, the presented results numerical applications testifies that the presented material is familiar to the candidate.

**Response and Corrections:**

1. The candidate do not familiar with the results of numerical applications, the examples are random selected. The examples are just to show the new theory is actually works in the real applications. Also in the paper, we have enough details to explain the theory how to run in the examples step by step, which do testified the theory is suitable for the examples we give.



**Professor Dr K Kolowrocki Comments:**

Concerning Section 3.6: The deficiency of this section on giving theorems (21, 36) without profs and references should be clarified to the COA.

**Contents in Examined Thesis:**

The theorems (21, 36) don't exist in the section 3.6

**Professor Dr K Kolowrocki Comments:**

| Concerning Section 3.7: The deficiency of this section on giving Propositions (9, 19) without profs and references should be clarified to the COA. |
|---|
| **Contents in Examined Thesis:** |
| Proposition 9: The relative accuracy-wise metric $d_v, v>0$ satisfies following three properties:<br>(1) For all non-negative reals r and s, $0 \le d_v(\cdot,\cdot) < 1$;<br>(2) For all non-negative reals r and s, $r \le s \le t$, then<br>$d_v(r,s) \le d_v(r,t)$,<br>$d_v(r,t) \le d_v(s,t)$<br>(3) For all non-negative reals *r* and *s*, $0 \le r \le s \le M$<br>$\frac{\lvert r-s \rvert}{v+2M} \le d_v(r,s) \le \frac{\lvert r-s \rvert}{v}$ |
| **Response and Corrections:** |
| **Proof:** $\because$ **Definition 8:** The relative accuracy-wise metric $d_v : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ is defined by<br>$d_v = \frac{\lvert r-s \rvert}{v+r+s}$ where the parameter $v \in (0,\infty)$.<br>**(1) Prof :For all non-negative reals r and s,** $0 \le d_v(\cdot,\cdot) < 1$**;**<br>$\because$ all non-negative reals r and s, $d_v, v>0$<br>1. if $r = s = 0$<br>$\because r+s = \lvert r-s \rvert = 0$<br>and $v > 0$<br>$\therefore \frac{\lvert r-s \rvert}{v+r+s} = 0$<br>$\therefore \frac{\lvert r-s \rvert}{v+r+s} = 0$<br>2. if $r>0, s=0$ or $r=0, s>0$<br>and $v>0$<br>$\therefore \frac{\lvert r-s \rvert}{r+s} = 1$<br>$\because v>0, r+s = \lvert r-s \rvert > 0$<br>$\therefore r+s+v > \lvert r-s \rvert > 0$<br>$\therefore 0 < \frac{\lvert r-s \rvert}{v+r+s} < 1$ |

3. if $r > 0, s > 0$

$\therefore r + s > |r - s| \geq 0$

$\because v > 0$

$\therefore v + r + s > |r - s| \geq 0$

$$\therefore 0 \leq \frac{|r-s|}{v+r+s} < 1$$

From above 1,2 and 3 , (1) is proved.

**（2） Proof: For all non-negative reals r and s, $r \leq s \leq t$, then**

$d_v(r,s) \leq d_v(r,t)$,

$d_v(r,t) \leq d_v(s,t)$

For all non-negative reals r and s, $r \leq s \leq t$

$$d_v(r,s) = \frac{|r-s|}{v+r+s}, d_v(r,t) = \frac{|r-t|}{v+r+t}$$

$\because 0 \leq r \leq s \leq t$

$\therefore 0 \geq -s \geq -t$

$\because r \geq 0$

$\therefore 0 \geq r - s \geq r - t$

$\therefore 0 \leq |r - s| \leq |r - t|$

$\because t - s \geq 0$

$\therefore |r - t| = |r - s| + (t - s)$

$$\therefore \frac{|r-t|}{v+r+t} = \frac{|r-s|+(t-s)}{(v+r+s)+(t-s)}$$

$$\begin{aligned}
&\frac{|r-t|}{v+r+t} - \frac{|r-s|}{v+r+s} \\
&= \frac{|r-t|(v+r+s) - |r-s|(v+r+t)}{(v+r+t)(v+r+s)} \\
&= \frac{(|r-s|+(t-s))(v+r+s) - |r-s|((v+r+s)+(t-s))}{(v+r+t)(v+r+s)} \\
&= \frac{(t-s)(v+r+s) - |r-s|(t-s)}{(v+r+t)(v+r+s)} = \frac{(t-s)((v+r+s)-|r-s|)}{(v+r+t)(v+r+s)}
\end{aligned}$$

According to (1)'s result, we have

$$0 \le \frac{|r-s|}{v+r+s} < 1$$

$$\therefore |r-s| < v+r+s$$

$$\therefore (v+r+s) - |r-s| > 0$$

$$\because t-s \ge 0 \text{ and } (v+r+t)(v+r+s) > 0$$

$$\therefore \frac{(t-s)\left((v+r+s)-|r-s|\right)}{(v+r+t)(v+r+s)} \ge 0$$

Hence that we have

$$\frac{|r-t|}{v+r+t} - \frac{|r-s|}{v+r+s} \ge 0$$

$$\frac{|r-s|}{v+r+s} \le \frac{|r-t|}{v+r+t}$$

$\therefore d_v(r,s) \le d_v(r,t)$ is verified

For all non-negative reals r and s, $r \le s \le t$

$$d_v(s,t) = \frac{|s-t|}{v+s+t}, d_v(r,t) = \frac{|r-t|}{v+r+t}$$

$$\because 0 \le r \le s \le t$$

$$\therefore 0 \ge -r \ge -s$$

$$\because t \ge 0$$

$$\therefore 0 \ge t-r \ge t-s$$

$$\therefore 0 \le |r-t| \le |s-t|$$

$$\because s-r \ge 0$$

$$\therefore |s-t| = |r-t| + (s-r)$$

$$\therefore \frac{|s-t|}{v+s+t} = \frac{|r-t| + (s-r)}{(v+r+t)+(s-r)}$$

$$\frac{|s-t|}{v+s+t}-\frac{|r-t|}{v+r+t}$$
$$=\frac{|s-t|(v+r+t)-|r-t|(v+s+t)}{(v+s+t)(v+r+t)}$$
$$=\frac{(|r-t|+(s-r))(v+r+t)-|r-t|((v+r+t)+(s-r))}{(v+s+t)(v+r+t)}$$
$$=\frac{(s-r)(v+r+t)-|r-t|(s-r)}{(v+s+t)(v+r+t)}=\frac{(s-r)((v+r+t)-|r-t|)}{(v+s+t)(v+r+t)}$$

According to (1)'s result, we have

$$0\le\frac{|r-t|}{v+r+t}<1$$
$$\therefore |r-t|<v+r+t$$
$$\therefore (v+r+t)-|r-t|>0$$

$\because s-r\ge 0$ and $(v+s+t)(v+r+t)>0$

$$\therefore \frac{(s-r)((v+r+t)-|r-t|)}{(v+s+t)(v+r+t)}\ge 0$$

Hence that we have

$$\frac{|s-t|}{v+s+t}-\frac{|r-t|}{v+r+t}\ge 0$$
$$\frac{|r-t|}{v+r+t}\le\frac{|s-t|}{v+s+t}$$

$\therefore d_v(r,t)\le d_v(s,t)$ is verified

From above 2 situations, (2) is proved.

（3） **Proof: For all non-negative reals r and s,** $0\le r\le s\le M$

$$\frac{|r-s|}{v+2M}\le d_v(r,s)\le\frac{|r-s|}{v}$$
$$\frac{|r-s|}{v+2M}\le\frac{|r-s|}{v+r+s}\le\frac{|r-s|}{v}$$

$$\frac{|r-s|}{v+r+s}-\frac{|r-s|}{v+2M}$$
$$=\frac{|r-s|(v+2M)-|r-s|(v+r+s)}{(v+r+s)(v+2M)}$$
$$=\frac{|r-s|(2M-r-s)}{(v+r+s)(v+2M)}$$
$$=\frac{|r-s|((M-r)+(M-s))}{(v+r+s)(v+2M)}$$

$\because 0 \le r \le s \le M$ **and** $d_v, v > 0$

$\therefore |r-s| \ge 0, ((M-r)+(M-s)) \ge 0, (v+r+s)(v+2M) > 0$

$$\therefore \frac{|r-s|((M-r)+(M-s))}{(v+r+s)(v+2M)} \ge 0$$

$\therefore \frac{|r-s|}{v+2M} \le d_v(r,s)$ is verified

$$\frac{|r-s|}{v}-\frac{|r-s|}{v+r+s}$$
$$=\frac{|r-s|(v+r+s)-|r-s|v}{v(v+r+s)}$$
$$=\frac{|r-s|(r+s)}{v(v+r+s)}$$

$\because 0 \le r \le s \le M$ **and** $d_v, v > 0$

$\therefore |r-s| \ge 0, (r+s) \ge 0, v(v+r+s) > 0$

$$\therefore \frac{|r-s|(r+s)}{v(v+r+s)} \ge 0$$

$\therefore d_v(r,s) \le \frac{|r-s|}{v}$ is verified

Hence that $\frac{|r-s|}{v+2M} \le d_v(r,s) \le \frac{|r-s|}{v}$, (3) is proved.

**Contents in Examined Thesis:**

Lemma 19: Let *f(x)* be a differentiable function with continuous derivative *f'(x)* at *x*. Then the forward Newton difference quotient approximation is

$$f'(x)=\frac{f(x+h)-f(x)}{h}+\varepsilon$$

Where the error term

$$E[\varepsilon_a]=O(h^{-1}), V[\varepsilon_a]=O(h^{-2})$$

| |
|---|
| The proof of the lemma is simply application of Taylor's expansion of a function. |
| **Response and Corrections:** |
| **Proof:**<br>$f'(x)=\frac{f(x+h)-f(x)}{h}+\varepsilon$<br>The derivatives of a function $f$ at a point $x$ provide polynomial approximations to that function near $x$. For example, if $f$ is twice differentiable, then<br>$f(x+h)=f(x)+f'(x)h+\frac{1}{2}f''(x)h^2+\frac{1}{6}f^{(3)}(x)h^3+\cdots$<br>$f(x+h)-f(x)=f'(x)h+\frac{1}{2}f''(x)h^2+\frac{1}{6}f^{(3)}(x)h^3+\cdots$<br>$f'(x)=\frac{f(x+h)-f(x)-\frac{1}{2}f''(x)h^2-\frac{1}{6}f^{(3)}(x)h^3-\cdots}{h}$<br>$f'(x)=\frac{f(x+h)-f(x)}{h}-\frac{1}{2}f''(x)h-\frac{1}{6}f^{(3)}(x)h^2-\cdots$<br>**Let** $\varepsilon=-\frac{1}{2}f''(x)h-\frac{1}{6}f^{(3)}(x)h^2-\cdots$<br>**Then we could reserve following results**<br>$E[\varepsilon_a]=O(h^{-1}),V[\varepsilon_a]=O(h^{-2})$<br>**The Lemma is proved.** |



| |
|---|
| **Professor Dr S Kar Comments:** |
| Dissertation is in the form of a book. Therefore, in the body of the thesis, it should not be mentioned like "In this paper,… …" in the Abstract page-3.22, 2nd line; page3-44, 2nd line, page 3-60 2nd line etc. |
| **Contents in Examined Thesis:** |
| "In this paper,… …" in the Abstract page-3.22: In this paper, we propose a new global optimization algorithm inspired by stochastic model and graph theory, which is named as lambda algorithm (LA).<br>Page3-44, 2nd line: In this paper, a new algorithm, named as Nash-lambda by merging Nash equilibrium solution and the lambda algorithm, is proposed. |

Page 3-60 2nd line: In this section we utilize the Rosenbrock's function as an example for introducing the lambda algorithm.

**Response and Corrections:**

The thesis is written in papers collection style, each paper in the thesis is originally from the published journals. So the selected paper still uses the words as past.

**Professor Dr S Kar Comments:**

In a book, figures are not repeated. In this dissertation, Fig. 3.1.3 and Fig. 3.2.6 are same. Similarly, Fig.3.1.1 and Fig.3.1.2 and Fig.3.2.2, figs at 3-67 and 3-161 etc. are the same.

**Contents in Examined Thesis:**

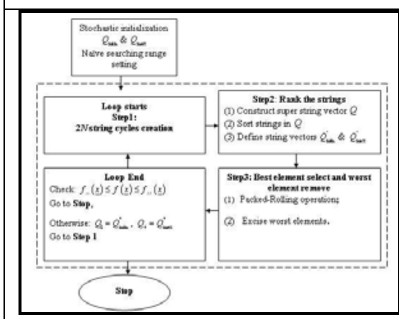



Figure 3.1.3 Flow chart of naïve five-element string algorithm

Figure 3.2.6 Flow chart of naïve five-element string algorithm

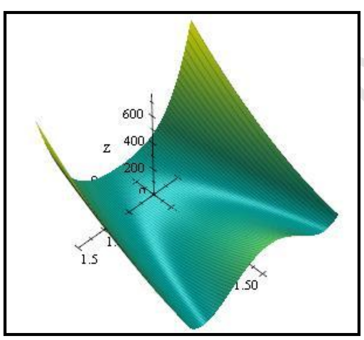



Figure 3.1.1 Plot of Rosenbrok function

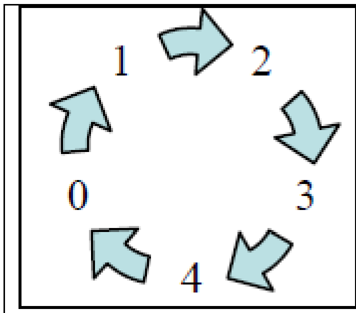



Figure 3.1.2 $\lambda[\ ]$ operation cycle

Figure 3.2.2 $\lambda[\ ]$ operation cycle

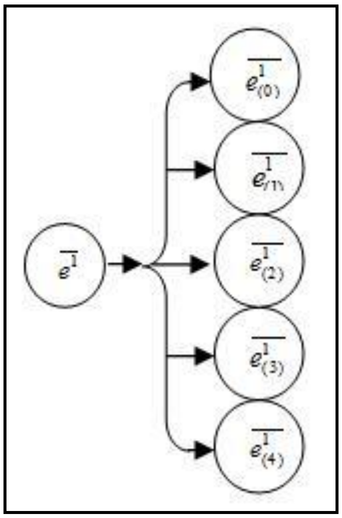



Figure 3.4.2 A string vector $\overline{e^1}$ and its cyclic vectors

Figure 3.7.4 A string vector $\overline{e^1}$ and its cyclic vectors

**Response and Corrections:**

The different papers published to different journals, and some of them talk about the same topic, i.e. lambda algorithm. It's very difficult for people who read one of the article but short of images which only possible in another paper. That's why for convenient, the different papers sometimes repeated some images in the book, it should be realized, the paper write in a papers collection style, the selected papers must honest to the original ones.

**Professor Dr S Kar Comments:**

English sentences at some places are meaningless i.e. construction are wrong. In chapter-5; conclusion; line 13 from below, "The third aim is to use new the knowledge" should be "… … the new knowledge", etc.

| **Contents in Examined Thesis:** |
|---|
| "The third aim is to use new the knowledge": The third aim is to use new the knowledge in developing uncertain canonical process regression to develop a probabilistic differential equations associate regression, which is a continue research of my master thesis "statistical-grey consistency of grey differential equation models". |
| **Response and Corrections:** |
| Corrected in the thesis<br>The third aim is to use the new knowledge in developing uncertain canonical process regression to develop a probabilistic differential equations associate regression, which is a continue research of my master thesis "statistical-grey consistency of grey differential equation models". |

| **Professor Dr S Kar Comments:** |
|---|
| Abbreviation of some terms like DEAR, MUP, etc. should have helped the readers etc. |
| **Contents in Examined Thesis:** |
| Accordingly, the general uncertainty decision-making approach in discrete and continues uncertainty environments including MUP Bayesian approach, are developed respectively.<br>More importantly, the new modelling family simplifies the DEAR model and make it being suitable to data mining task. At this point, DEAR modelling family can facilitate the statistical modelling selections such that the new statistical machine learning algorithm is in shape. |
| **Response and Corrections:** |
| Corrected in the thesis.<br>The overall aim of the research is to create a new statistical machine learning algorithm, named as Differential Equations Associated Regression Modelling (DEAR)-λ-algorithm.<br>Accordingly, the general uncertainty decision - making approach in discrete and continuous uncertainty environments including Maximum Uncertainty Principle (MUP) Bayesian approach, are developed respectively. |

# Appendix B: Lambda algorithm Matlab code

```
function  [Bestfitness,variables,TI,Ge]=lambda
name=input('Please enter the function name(must enter):example @function name\n');
Size=input('Please enter population size you want(could be empty):default 100\n');
if isempty(Size)
Size=100;
end
G=input('Please enter looping times you want(could be empty):default 100\n');
if isempty(G)
G=100;
end
 Codel=input('Please enter String length of each variables(could be empty):default 4\n');
if isempty(Codel)
Codel=4;
end
umaxo=input('please enter upper bound of variables you want(could be empty):default 10^6\n');
 if isempty(umaxo)
umaxo=10^6;
end
 umino=input('please enter lower bound of variable you want(could be empty):default -10^6\n');
if isempty(umino)
umino=-10^6;
end
n=input('Please enter number of variables(must enter):example 2\n');
Po=input('Please enter Probability measure to control the Process:example 0.5\n');
if isempty(Po)
Po=1.0;
end
mm=cell(1,n);
mmb=cell(1,n);
Q=cell(1,5);
Q{1,1}=round(4*rand(Size,n*Codel));
EX=round(4*rand(Size,n*Codel));
EX1=round(4*rand(Size,n*Codel));
for v=1:1:n
umax(v)=umaxo;umin(v)=umino;
end
Ge=0;
tic;
for k=1:1:G
time(k)=k;
for i=1:1:Size
for j=1:1:n*Codel
if Q{1,1}(i,j)==0
Q{1,2}(i,j)=1;Q{1,3}(i,j)=2;Q{1,4}(i,j)=3;Q{1,5}(i,j)=4;
```

```
elseif Q{1,1}(i,j)==1
Q{1,2}(i,j)=2;Q{1,3}(i,j)=3;Q{1,4}(i,j)=4;Q{1,5}(i,j)=0;
elseif Q{1,1}(i,j)==2
Q{1,2}(i,j)=3;Q{1,3}(i,j)=4;Q{1,4}(i,j)=0;Q{1,5}(i,j)=1;
elseif Q{1,1}(i,j)==3
Q{1,2}(i,j)=4;Q{1,3}(i,j)=0;Q{1,4}(i,j)=1;Q{1,5}(i,j)=2;
elseif Q{1,1}(i,j)==4
Q{1,2}(i,j)=0;Q{1,3}(i,j)=1;Q{1,4}(i,j)=2;Q{1,5}(i,j)=3;
end
 if  EX(i,j)==0
Q{1,6}(i,j)=1;Q{1,7}(i,j)=2;Q{1,8}(i,j)=3;Q{1,9}(i,j)=4;
elseif EX(i,j)==1
Q{1,6}(i,j)=2;Q{1,7}(i,j)=3;Q{1,8}(i,j)=4;Q{1,9}(i,j)=0;
elseif EX(i,j)==2
Q{1,6}(i,j)=3;Q{1,7}(i,j)=4;Q{1,8}(i,j)=0;Q{1,9}(i,j)=1;
elseif EX(i,j)==3
Q{1,6}(i,j)=4;Q{1,7}(i,j)=0;Q{1,8}(i,j)=1;Q{1,9}(i,j)=2;
elseif EX(i,j)==4
Q{1,6}(i,j)=0;Q{1,7}(i,j)=1;Q{1,8}(i,j)=2;Q{1,9}(i,j)=3;
end
if  EX1(i,j)==0
S{1,6}(i,j)=1;S{1,7}(i,j)=2;S{1,8}(i,j)=3;S{1,9}(i,j)=4;
elseif EX1(i,j)==1
S{1,6}(i,j)=2;S{1,7}(i,j)=3;S{1,8}(i,j)=4;S{1,9}(i,j)=0;
elseif EX1(i,j)==2
S{1,6}(i,j)=3;S{1,7}(i,j)=4;S{1,8}(i,j)=0;S{1,9}(i,j)=1;
elseif EX1(i,j)==3
S{1,6}(i,j)=4;S{1,7}(i,j)=0;S{1,8}(i,j)=1;S{1,9}(i,j)=2;
elseif EX1(i,j)==4
S{1,6}(i,j)=0;S{1,7}(i,j)=1;S{1,8}(i,j)=2;S{1,9}(i,j)=3;
end
end
end
E=[Q{1,1};Q{1,2};Q{1,3};Q{1,4};Q{1,5};Q{1,6};Q{1,7};Q{1,8};Q{1,9};EX;EX1;S{1,6};S{1,7
};S{1,8};S{1,9}];
for s=1:1:15*Size
m=E(s,:);
for v=1:1:n
y(v)=0;
mm{1,v}=m(Codel*(v-1)+1:1:v*Codel);
for i=1:1:Codel
y(v)=y(v)+mm{1,v}(i)*5^(Codel-i);
end
x(v)=(umax(v)-umin(v))*y(v)/(5^Codel)+umin(v);
r(1,v)=x(v);
end
F(s)=name(r);
end
```

```
Ji=1./F;
BestJ(k)=max(Ji);
fi=F;
 [Oderfi,Indexfi]=sort(fi);
 [Oderfi1,Indexfi1]=sort(fi,'descend');
 Bestfitness=Oderfi(1);
for i=1:1:Size
TempE(i,:)=E(Indexfi(i),:);
TempE2(i,:)=E(Indexfi1(i),:);
end
BestS=TempE(1,:);
bfi(k)=Bestfitness;
BS(k,:)=BestS;
for j=1:Codel:n*Codel
 xx=TempE(:,j);
for i=1:1:Size
if xx(i)~=xx(1)
break
end
end
y0=0;y1=0;y2=0;y3=0;y4=0;
if xx(1)==0
y0=(i-1)/Size;
elseif xx(1)==1
y1=(i-1)/Size;
elseif xx(1)==2
y2=(i-1)/Size;
elseif xx(1)==3
y3=(i-1)/Size;
elseif xx(1)==4
y4=(i-1)/Size;
end
v=fix(j/Codel)+1;umax1(v)=umax(v);umin1(v)=umin(v);
if y0>Po && TempE(1,j)==0
umax(v)=(umax1(v)-umin1(v))/5+umin1(v);
umin(v)=umin1(v);
end
if y1>Po && TempE(1,j)==1
umax(v)=(umax1(v)-umin1(v))*2/5+umin1(v);
umin(v)=(umax1(v)-umin1(v))/5+umin1(v);
end
if y2>Po && TempE(1,j)==2
umax(v)=(umax1(v)-umin1(v))*3/5+umin1(v);
umin(v)=(umax1(v)-umin1(v))*2/5+umin1(v);
end
if y3>Po && TempE(1,j)==3
umax(v)=(umax1(v)-umin1(v))*4/5+umin1(v);
umin(v)=(umax1(v)-umin1(v))*3/5+umin1(v);
```

```
end
if y4>Po && TempE(1,j)==4
umin(v)=(umax1(v)-umin1(v))*4/5+umin1(v);
umax(v)=umax1(v);
end
if umax(v)~=umax1(v) || umin(v)~=umin1(v)
QBack=TempE(:,(j+1):v*Codel);
TempE(:,j:(v*Codel-1))=QBack;
TempE(:,v*Codel)=round(4*rand(Size,1));
TempE(1,v*Codel)=0;
end
end
TempE1=TempE;
for i=1:1:Size-2
for j=1:1:n*Codel
if TempE(i,j)==TempE(i+1,j) && TempE(i,j)~=4
TempE(i+1,j)=TempE(i+1,j)+1;
elseif TempE(i,j)==TempE(i+1,j) && TempE(i,j)==4
TempE(i+1,j)=0;
elseif TempE(i,j)==TempE(i+2,j) && TempE(i,j)~=4
TempE(i+2,j)=TempE(i+2,j)+1;
elseif TempE(i,j)==TempE(i+2,j) && TempE(i,j)==4
TempE(i+2,j)=0;
end
if TempE2(i,j)==TempE2(i+1,j) && TempE2(i,j)~=4
TempE2(i+1,j)=TempE2(i+1,j)+1;
elseif TempE2(i,j)==TempE2(i+1,j) && TempE2(i,j)==4
TempE2(i+1,j)=0;
elseif TempE2(i,j)==TempE2(i+2,j) && TempE2(i,j)~=4
TempE2(i+2,j)=TempE2(i+2,j)+1;
elseif TempE2(i,j)==TempE2(i+2,j) && TempE2(i,j)==4
TempE2(i+1,j)=0;
End
if TempE1(i,j)==TempE1(i+1,j) && TempE1(i,j)~=0
TempE1(i+1,j)=TempE1(i+1,j)-1;
elseif TempE1(i,j)==TempE1(i+1,j) && TempE1(i,j)==0
TempE1(i+1,j)=4;
elseif TempE1(i,j)==TempE1(i+2,j) && TempE1(i,j)~=0
TempE1(i+2,j)=TempE1(i+2,j)-1;
elseif TempE1(i,j)==TempE1(i+2,j) && TempE1(i,j)==0
TempE1(i+2,j)=4;
end
end
end
for i=1:1:Size-1
for j=1:1:n*Codel
if TempE2(1,j)==TempE(i+1,j)
TempE(i+1,j)=TempE(1,j);
```

```
end
if TempE(1,j)==TempE2(i+1,j)
TempE2(i+1,j)=TempE2(1,j);
end
end
end
TempE1(1,:)=round(4*rand(1,n*Codel));
Q{1,1}=TempE;
EX=TempE1;
EX1=TempE2;
for v=1:1:n
yb(v)=0;
mmb{1,v}=BestS(Codel*(v-1)+1:1:v*Codel);
for i=1:1:Codel
yb(v)=yb(v)+mmb{1,v}(i)*5^(Codel-i);
end
variables(v)=(umax(v)-umin(v))*yb(v)/(5^Codel)+umin(v);
end
figure(1);
subplot(2,1,1);
set(gcf,'CurrentAxes',gca)
plot(time,BestJ);
xlabel('Times');ylabel('Inverse meaure of best fitness value');
subplot(2,1,2);
set(gcf,'CurrentAxes',gca)
plot(time,bfi)
xlabel('Times');ylabel('Best optimization value');
if k>2 &&  bfi(k)~=bfi(k-1)
TI=toc;Ge=k;
end
end
```

# References


1. Abe, S. (2005). Support vector machines for pattern classification. Springer-Verlag, New York.
2. Abe, S. and Inoue, T. (2001). Fast training of support vector machines by extracting boundary data. Artif Neural Netw ICANN 2130:308–313.
3. Allen, F.R., Ambikairajah, E., Lovell N.H. and Celler, B.G. (2006). Cl assification of a known sequence of motions and postures from accelerometry data using adapted Gaussian mixture models. Physiol Meas 27(10):935–952.
4. Bao, L. and Intille, S.S. (2004). Activity recognition from user-annotated acceleration data. Pervasive Comput 3001:1–17.
5. Ben-Gal, I. (2007). Bayesian Networks, in Ruggeri F., Faltin F. and Kenett R., Encyclopedia of Statistics in Quality & Reliability. England: John Wiley & Sons Ltd.
6. Benson, C.T. and Losey, N.E. (1971). On a Graph of Hoffman and Singleton. J. Combin. Th. Ser. B 11: 67-79.
7. Bernardo, J.M. and Smith, A.F.M. (1994). Bayesian Theory. England: John Wiley & Sons Ltd.
8. Billard L, Diday E (2000) Regression Analysis for Interval-Valued Data, In: Data Analysis, Classification and Related Methods, Proceedings of the 7th Conference of the International Federation of Classification Societies, Springer-Verlag, Belgium, pp 369–374.
9. Blische, W.R. and Murthy, D.N.P. (2000). Reliability-Modeling, Prediction, and Optimization. New York: John Wiley & Sons, Inc.
10. Bundy, A. (1985). Incidence calculus: a mechanism for probabilistic reasoning. Journal of Automated Reasoning 1:263–283.
11. Bundy, A. (1986). Correctness criteria of some algorithms for uncertain reasoning using incidence calculus. Journal of Automated Reasoning 2:109–126.
12. Buntine, W. (1994). Operations for learning with graphical model. Journal of Artificial Intelligence Research 2.
13. Carvalho, H. and Machado, V.C. (2006). Fuzzy set theory to establish resilient production systems. Proc. Of IIE Annual Conference and Exhibition.

14. Chen, X.R. (1981). Introductory Mathematical Statistical Theory. The Scientific Publishing House, Beijing, China.
15. Crosby, J.L. (1973). Computer Simulation in Genetics. London: John Wiley & Sons.
16. Cucchiara, R., Grana, C., Prati, A. and Vezzani, R. (2005). Probabilistic posture classification for human-behavior analysis. IEEE Trans Syst Man Cybern Part A 35(1):42–54.
17. Cui, Y.H. (2007). Statistical-Grey Consistent Grey Differential Equation Modelling. MSc thesis, University of Cape Town, South Africa.
18. Cui, Y.H. and Guo, R. (2008). Harmony element algorithm-A naïve initial searching range, Proceedings of the Internatioanl Conference on 'Advances in Mechanical Engineering', pp 479-484.
19. Cui, Y.H. and Guo, R. (2011). Nash-lambda Algorithm with Applications in Reliability and Risk. Journal of Polish Safety and Reliability Association, Summer Safety & Reliability Seminars 1:51–58.
20. Cui, Y.H. and Guo, R. (2011). Lambda Algorithm and Nash Numerical Solution**.** In the Proceedings of the Second International Conference on Uncertainty Theory (ICUT2011 http://orsc.edu.cn/icut), August 6-11, 2011, Lhasa, Tibet, China, pp 86-96, Editors: Kaoru Hirota, Dan A. Ralescu, Jin Peng. ISSN 2079-5238. Copyright © 2011 by International Consortium for Uncertainty Theory.
21. Cui, Y.H., Guo, R, Dunne, T. and Guo, D. (2010). Decision Theory under General Uncertainty. Journal of Polish Safety and Reliability Association, Summer Safety & Reliability Seminars 1:51–66.
22. Cui, Y.H., Guo, R., Dunne, T. and Guo, D. (2011). Bayesian Uncertainty Decision Analysis. Reliability & Risk Analysis: Theory & Applications, Electronic Journal of International Group on Reliability – Gnedenko E-Forum 2(1):70–81.
23. Cui Y.H., Guo R. and Guo, D. (2009). A Naïve Five-Element String Algorithm. Journal of Software 4(9):925–934.
24. Cui, Y.H., Guo, R. and Guo, D. (2010). Lambda Algorithm. Journal of Uncertain Systems 4(1):22–33, World Academic Union.
25. Cui, Y.H., Guo, R. and Guo, D. (2011). Lambda Algorithm and Maximum Likelihood Estimation. Journal of Polish Safety and Reliability Association, Summer Safety & Reliability Seminars 1:59–72.

26. Cui, Y.H., Guo, R. and Guo, D. (2011). Theory of λ algorithm**.** In the Proceedings of the Ninth China Annual Conference on Uncertainty, jointly with the Fifth Chinese Artificial Intelligent Computation Conference & the thirteenth Chinese Youth Information and Management Science Conference, July 27-31, 2011, Nanjing, China, pp 305-320, ISBN: 962-8266-65-2, Editor: Yanguo Zhu, Xiangqun, Yang, Jing Peng, and Yufu Ning. Publisher, Global-link Publisher, Hong Kong.
27. Cui, Y.H., Guo, R., Savsani, V.J., Rao, R.V. and Vakharia, D.P. (2008). Harmony Element Algorithm – A Naive Initial Searching Range. Proceedings of the International Conference on 'Advances in Mechanical Engineering', December 15-17, 2008, S.V. National Institute of Technology, Surat – 395 007, Gujarat, India, pp 479-484, Editor: R.V. Rao, Publisher: SVNIT, Surat, India.
28. Dai, W., Guo, R., Guo, D. and Cui, Y.H. (2011). The Autocorrelation of a Standard Uncertain Canonical Process**.** In the Proceedings of the Second International Conference on Uncertainty Theory (ICUT2011 http://orsc.edu.cn/icut), August 6-11, 2011, Lhasa, Tibet, China, pp 6-12, Editors: Kaoru Hirota, Dan A. Ralescu, Jin Peng. ISSN 2079-5238. Copyright © 2011 by International Consortium for Uncertainty Theory.
29. DeGroot, M.H. (1970). Optimal Statistical Decisions. New York: McGraw-Hill, Inc.
30. Deng, J.L. (1984). Grey dynamic modeling and its application in long-term prediction of food productions. Exploration of Nature 3(3):7–43.
31. Deng, J.L. (1985). Grey Systems (Social·Economical). The Publishing House of Defense Industry, Beijing.
32. Deng, J.L. (2002). Grey Prediction and Decision, 2nd edn. Huazhong University of Technology Press, Wuhan, China.
33. Deng, J.L. (2002). Grey Theoretical Foundations. The Publishing House of Hua-Zhong University of Science and Technology, Wuhan, China.
34. Deng, J.L. (2004). The Primary Methods of Grey System Theory (A Chinese-English Terminology Dictionary), 2nd edn. Huazhong University of Technology Press, Wuhan, China.
35. Domingos, P. and Pazzani, M. (1997). On the the optimality of the simple Bayesian classifier under zero-one loss. Mach Learn 29(2–3):103–130.
36. Draper, N. and Smith, H. (1966). Applied Regression Analysis, 2nd edn. John Wiley & Sons Inc, New York, USA.

37. Ferguson, T.S. (1967). Mathematical Statistics: A Decision Theoretic Approach.New York: Academic Press Inc.
38. Fraser, A. (1957). Simulation of genetic systems by automatic digital computers. I. Introduction. Aust. J. Biol. Sci. 10: 484-491.
39. Fraser, A. and Donald, B. (1970). Computer Models in Genetics. NewYork: McGraw-Hill.
40. Guo, D., Guo, R., Midgley, G.F. and Ebrahim, I. (2007). DEMR Modeling Brunsvigia Litoralis Plant Distribution in 2004, Western Region District Council, South Africa, Proceedings of the Sixth International Conference on Information and Management Sciences, Lhsa, Tibet, China, pp 770–782.
41. Guo, D., Guo, R., Cui, Y.H., Midgley, G.F., Altwegg, R., and Thiart, C. (2011). Climate Change Impact on Quiver Trees in Arid Namibia**.** In the Proceedings of the Second International Conference on Uncertainty Theory (ICUT2011 http://orsc.edu.cn/icut), August 6-11, 2011, Lhasa, Tibet, China, pp 187-194, Editors: Kaoru Hirota, Dan A. Ralescu, Jin Peng. ISSN 2079-5238. Copyright © 2011 by International Consortium for Uncertainty Theory.
42. Guo, D., Guo, R. and Cui, Y.H. (2011). Application of Uncertain Canonical Process in Spatial Predictions**.** In the Proceedings of the Ninth China Annual Conference on Uncertainty, jointly with the Fifth Chinese Artificial Intelligent Computation Conference & the thirteenth Chinese Youth Information and Management Science Conference, July 27-31, 2011, Nanjing, China, pp 305-320, ISBN: 962-8266-65-2, Editor: Yanguo Zhu, Xiangqun, Yang, Jing Peng, and Yufu Ning. Publisher, Global-link Publisher, Hong Kong.
43. Guo, D., Guo, R., Cui, Y.H. and Dunne, T. (2011). The Uncertain Fractional Canonical Process Regression**.** In the Proceedings of the Second International Conference on Uncertainty Theory (ICUT2011 http://orsc.edu.cn/icut), August 6-11, 2011, Lhasa, Tibet, China, pp 199-207, Editors: Kaoru Hirota, Dan A. Ralescu, Jin Peng. ISSN 2079-5238. Copyright © 2011 by International Consortium for Uncertainty Theory.
44. Guo, D., Guo, R. and Thiart, C. (2006). The Coupling of regression Modeling and Differential Equation Model in GM(1,1) Modeling and Extended GM(1,1) Models. Journal of Grey System 9(2):143–154.
45. Guo, D. Guo, R., Thiart, C. and Cui, Y.H. (2011). Imprecise Uncertainty Modeling of Air Pollutant $PM_{10}$**.** In the Proceedings of the Second International Conference on Uncertainty Theory (ICUT2011 http://orsc.edu.cn/icut), August 6-11, 2011, Lhasa, Tibet, China, pp 175-

181, Editors: Kaoru Hirota, Dan A. Ralescu, Jin Peng. ISSN 2079-5238. Copyright © 2011 by International Consortium for Uncertainty Theory.

46. Guo, R. (2005). Repairable System Modeling via Grey Differential Equations. Journal of Grey System 8(1): 69–92.
47. Guo, R. (2007). Grey Differential Equation GM(1,1) Model in Reliability Engineering. Computational Intelligence in Reliability Engineering – Vol. 2. New Metaheuristics, Neural and Fuzzy Techniques in Reliability, Chapter 26, pp 387–413.
48. Guo, R. (2011). Bayesian Reliability Modelling. In: International Encyclopedia of Statistical Science, Lovric M (ed), pp 104-105, Springer.
49. Guo, R., Cui, Y.H. and Guo, D. (2010). Uncertainty Linear Regression models. (Draft paper submitted to Journal of uncertain systems).
50. Guo, R., Cui, Y.H. and Guo, D. (2011). Uncertain Canonical Process Driven Models and Data Analysis**.** In the Proceedings of the Ninth China Annual Conference on Uncertainty, jointly with the Fifth Chinese Artificial Intelligent Computation Conference & the thirteenth Chinese Youth Information and Management Science Conference, July 27-31, 2011, Nanjing, China, pp 429-443, ISBN: 962-8266-65-2, Editor: Yanguo Zhu, Xiangqun, Yang, Jing Peng, and Yufu Ning. Publisher, Global-link Publisher, Hong Kong.
51. Guo, R., Dai, W., Guo, D., Cui, Y.H. and Dunne, T. (2011). Uncertain Canonical Process Regression Models**.** In the Proceedings of the Second International Conference on Uncertainty Theory (ICUT2011 http://orsc.edu.cn/icut), August 6-11, 2011, Lhasa, Tibet, China, pp 36-44, Editors: Kaoru Hirota, Dan A. Ralescu, Jin Peng. ISSN 2079-5238. Copyright © 2011 by International Consortium for Uncertainty Theory.
52. Guo, R., Dunne, T. and Cheng, C.Y. (2008). DEAR CUSUM Control Chart, Proceedings of 3rd Asian International Workshop on Advanced Reliability Modelling III: Global Aspect of Reliability and Maintainability, Taichung, Taiwan, pp 800–808.
53. Guo, R. and Guo, D. (2007). Univariate DEMR Model – Formation and Foundation, Proceedings of 2007 IEEE International Conference on Grey Systems and Intelligent Services, Nanjing China, pp 13–18.
54. Guo, R. and Guo, D. (2007). Multivariate DEMR Model – Formation and Foundation, Proceedings of 2007 IEEE International Conference on Grey Systems and Intelligent Services, Nanjing China, pp 19-25.

55. Guo, R. and Guo, D. (2009). DEAR Theory in System Analysis. Reliability & Risk Analysis: Theory & Applications, Electronic Journal of International Group on Reliability – Gnedenko E-Forum 2(1):64–78.

56. Guo, R. and Guo, D. (2009). Random Fuzzy Variable Foundation for Grey Differential Equation Modelling. Soft Computing 13(2):185–201.

57. Guo, R., Guo, D. and Cui, Y.H. (2010). Wave-like bathtub hazard function. Proc. of the Ninth International Conference on Information and Management Sciences, 434-440.

58. Guo, R., Guo, D., Cui, Y.H. and Dai, W. (2011). Uncertain Temporal Regression Models**.** In the Proceedings of the Second International Conference on Uncertainty Theory (ICUT2011 http://orsc.edu.cn/icut), August 6-11, 2011, Lhasa, Tibet, China, pp 73-79, Editors: Kaoru Hirota, Dan A. Ralescu, Jin Peng. ISSN 2079-5238. Copyright © 2011 by International Consortium for Uncertainty Theory.

59. Guo, R. and Guo, D. and Cui, Y.H. (2011). Uncertain DRAR Models**.** In the Proceedings of the Second International Conference on Uncertainty Theory (ICUT2011 http://orsc.edu.cn/icut), August 6-11, 2011, Lhasa, Tibet, China, pp 104-111, Editors: Kaoru Hirota, Dan A. Ralescu, Jin Peng. ISSN 2079-5238. Copyright © 2011 by International Consortium for Uncertainty Theory.

60. Guo, R. and Guo, D. and Cui, Y.H. (2011). The Autocorrelation of an Uncertain Integral Driven Process**.** In the Proceedings of the Second International Conference on Uncertainty Theory (ICUT2011 http://orsc.edu.cn/icut), August 6-11, 2011, Lhasa, Tibet, China, pp 104-111, Editors: Kaoru Hirota, Dan A. Ralescu, Jin Peng. ISSN 2079-5238. Copyright © 2011 by International Consortium for Uncertainty Theory.

61. Guo, R. and Guo, D. and Cui, Y.H. (2011). An Uncertain Regression Model. Proceedings of IEEE International Conference on Grey Systems and Intelligent Services, September 15-17, Nanjing, China, 2011, (accepted). IEEE Catalog Number: 07EX1882C, ISBN: 1-4244-1294-3, Library of Congress: 2007926987, Editor-in-Chief: Sifeng Liu (ISTP-indexed) (ISTP-indexed)

62. Guo, R., Guo, D., Dunne, T. and Thiart, C. (2009). DEAR Model – The Theoretical Foundation, Journal of Uncertain Systems 3(1):36–51.

63. Guo, R., Guo, D., Midgley, G.F. and Rebelo, A.G. (2007). Bivariate DEMR Modeling of Range Shifts in Aulax Umbellata Shrubs From 1992 to 2002 in South Africa, Proceedings of

the Sixth International Conference on Information and Management Sciences, Lhsa, Tibet, China, pp 783–791.

64. Guo, R., Guo, D, Thiart, C. (2007). DEMR Embedded Models in Density Saddlepoint Approximations, Proceedings of IEEE International Conference on Grey Systems and Intelligent Services, Nanjing, China, pp 57–62.
65. Guo, R. and Love, C.E. (2005). Grey Repairable System Analysis (Plenary Lecture), Advances in Safety and Reliability, Proceedings of the European Safety and Reliability Conference, Tri City, Poland, pp 753–766.
66. Guo, R., Love, C.E. and Cui, Y.H., (2010). Maximum Likelihood Estimation of 2-Parameter Wave-Like Lifetime Distributions. Advanced Reliability Modeling IV-Beyond the Tradidtional Reliability and Maintainability Approaches. Proc. of the 4th Asia-Pacific International Symposium, 225-232.
67. Guo, R., Thiart, C., Cui, Y.H. and Guo, D. (2011). Lifetime Distributions with Wave-like Bathtub hazard. Journal of Polish Safety and Reliability Association, Summer Safety & Reliability Seminars, SSARS 2011, pp 115 – 122. Vol. 1: ISBN: 978-83-925436-2-6. Editors in Chief: Krzysztof Kolowrocki, & Joanna Soszynska-Budny.
68. Haussler, D. (1990). Probably approximately correct learning. In AAAI-90 Proceedings of the Eight National Conference on Artificial Intelligence, Boston, MA, pp 1101–1108, American Association for Artificial Intelligence.
69. Haussler, D. (1992). Decision Theoretic Generalizations of the PAC Model for Neural Net and Other Learning Applications. Information and Computation 100:78–150.
70. Heymann, P.B. (2003). Dealing with Terrorism after Sptember 11, 2001: An Overview. In: Howitt, A., Pangi, R., (eds.). Preparing for Domestic Terrorism. MIT Press, Cambridge, MA, 57-72.
71. Hsu, C.W. and Lin, C.J. (2002). A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13(2):415–425.
72. Jackson, C. and Pascual, R. (2008). Optimal maintenance service contract negotiation with aging equipments. European Journal of Operational Research 189: 387-398.
73. Jin, A., Jiang, J.H., Lou, J.G. and Zhang, R. (2009). Software Reliability Modeling Based on Grey System Theory. Journal of Computer Applications 29(3):690– 694.

74. Kaplan, E.H., Mintz, A., Mishal, S. and Samban, C. (2005). What Happened to Suicide Bombings in Israel? Insights from a Terror Stock Model. Studies in Conflict & Terrorism 28:225-235.

75. Kearns, M. and Vazirani, U. (1994). An Introduction to Computational Learning Theory. MIT Press.

76. Keohane, N.O, and Zeckhauser, R. (2003). The Ecology of Terror Defense. Journal of Risk and Uncertainty 26: 201-229.

77. Kjrulff, U. (1992). A computational scheme for reasoning in dynamic probabilistic networks. Proc. of the Eighth Conference on Uncertainty in Artificial Intelligence, 121-129.

78. Koggalage, R. and Halgamuge, S. (2004). Reducing the number of training samples for fast support vector machine classification. Neural Inf Process Lett Rev 2(3):57–65.

79. Kolmogorov, A.N. (1950). Foundations of the Theory of Probability. Publisher: Chelsea, New York: Translated by Nathan Morrison.

80. Laffery, J. and Wasserman, L. (2006). Challenges in Statistical Machine Learning. Statistica Sinica Vol. 16: pp 307–322.

81. Lawless, J.F. (1982). Statistical Models and Methods for Lifetime Data. New York: Wiley.

82. Lee, P.M. (1989). Bayesian Statistics: An Introduction. New York: Oxford University Press.

83. Leslie, G. and Valiant, A. (1984). Theory of the Learnable. Communications of the ACM 27(11):1134–1142.

84. Liu, B.D. (1998). Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms. Computers Math. Applic. 36(7): 79-89.

85. Liu, B.D. (2007). Uncertainty Theory: An Introduction to Its Axiomatic Foundations. 2nd Edition; Berlin: Springer-Verlag Heidelberg.

86. Liu, B.D. (2009). Some research problems in uncertainty theory, Journal of Uncertain Systems, Vol.3, No.1, 3-10.

87. Liu, B.D. (2010). Uncertainty Theory. 3nd Edition. Berlin: Springer.

88. Liu, B.D. (2010). Uncertainty Theory: A Branch of Mathematics of Modelling Human Uncertainty. Springer-Verlag, Berlin.

89. Liu, S.F. and Lin, Y. (2006). Grey Information. Springer-Verlag, London.

90. Liu, Y.H. (2008). How to generate uncertain measure. Proceedings of Tenth National Youth Conference on Information and Management Sciences, August 3-7, Luoyang, pp23-26.

91. MacAllester, D. (1999). Some PAC-Bayesian theorems. Machine Learning 37:355–363.

92. Manbachi, M., Mahdloo, F. and Haghifam, M-R. (2010). A New Solution for Maintenance Scheduling in Deregulated Environment based on Lost Opportunity Cost of Market Participation and Reliability. Proc. of Modern Electric Power Systems 2010, Wroclaw, Poland, MEPS'10 -paper P4.
93. Masatoshi, S. and Ichiro, N. (2009). Cooperative and Noncooperative Multi-Level Programming. Springer.Dordrecht - Heidelberg - London - New York: Springer.
94. Mathworks Corp. (2008). Matlab Help Example: Rastrigin's function: Getting Started with Genetic algorithm (Genetic Algorithm and Direct Search Toolbox).
95. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M. and Euler, T. (2006). YALE: rapid prototyping for complex data mining tasks, Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 935–940.
96. Moore, R.E. (1966). Interval Analysis. Prentice-Hall, Englewood Cliff, NJ.
97. Myers, R.H. (2000). Classical and Modern Regression with Applications, 2nd edn. Duxbury Press.
98. Peng Z.X. and Iwamura, K. (2009). A sufficient and necessary condition of uncertainty distribution. http://orsc.edu.cn/online/090305.pdf
99. Primas, H. (1999). Basic Elements and Problems of Probability Theory. Journal of Scientific Exploration, 13(4): 579-613.
100. Rao, C.R. (1973). Linear Statistical Inference and Its Applications. John Wiley & Sons Inc, New York, USA.
101. Rasmussen, C.E. and Williams, C.K.I. (2006). Gaussian Processes for Machine Learning. MIT Press.
102. Sandler, T. and Hartley, K. (2001). Economics of alliances: The lessons for collective action. Journal of Economic Literature 34: 869-896.
103. Savsani, V.J., Rao, R.V., Vakharia, D.P., Cui, Y.H. and Guo, R. (2008). Modified Harmony Elements Algorithm for the Optimization of Mechanical Engineering Design. Proceedings of the International Conference on 'Advances in Mechanical Engineering', December 15-17, 2008, S.V. National Institute of Technology, Surat – 395 007, Gujarat, India, pp 545-550, Editor: R.V. Rao, Publisher: SVNIT, Surat, India.
104. Sefrioui, M. and Perlaux, J. (2000). Nash genetic algorithms: examples and applications. Proc. of the 2000 Congress on Evolutionary Computation 1: 509-516.

105. Simaan, M. and Cruz, J.B. (1973). A Stackelberg solution for games with many players. IEEE transactions on Automatic Control AC.18: 322-324.
106. Simaan, M. and Cruz, J.B. (1973). On the Stackelberg strategy in nonzero-sum games. Journal of Optimization Theory and App. 11: 533-555.
107. Snelson, E., Ghahramani, Z. and Rasmussen, C. (2004). Warped Gaussian processes. In: Thrun S, Saul L, Schölkopf B (ed), Advances in Neural Information Processing Systems Volume 16, pp 337–344.
108. Spata, M.O. and Rinaudo, S. (2010). Merging Nash Equilibrium Solution with Genetic Algorithm: The Game Genetic Algorithm. Journal of Convergence Information Technology. 5(9): 9-15.
109. Ushakov, I.A. (1994). Handbook of Reliability Engineering. New York.: John Wiley & Sons, Inc.
110. Varian, H.R. (1994). Sequential provision of public goods. Journal of Public Economics. 53: 165-186.
111. Wen, K.L. (2004). Grey Systems - Modeling and prediction. Yang's Scientific Press, Tucson, USA.
112. Weise, T. (2008). Global Optimization Algorithms:-Theory and Applications, 2nd Edition. http://it-weise.de/
113. Wikipedia, 2011. http://en.wikipedia.org/wiki/Bayesian_network
http://en.wikipedia.org/wiki/Bilevel_program
http://en.wikipedia.org/wiki/Game_theory
http://en.wikipedia.org/wiki/Multilevel_programming
http://en.wikipedia.org/wiki/Nash_equilibrim
http://en.wikipedia.org/wiki/Stackelberg_competition
114. Williams, C.K.I. and Rasmussen, C.E. (1996). Gaussian processes for regression. In: Touretzky DS, Mozer MC, Hasselmo, ME (ed), Advances in Neural Information Processing Systems Volume 8, pp 598–604, MIT Press.
115. Witten, I.H. and Frank, E. (2005). Data mining: practical machine learning tools and techniques, 2nd edn. The Morgan Kaufmann series in data management system, Elsevier.
116. Zadeh, L.A. (1965). Fuzzy Sets, Information and Control, Vol. 8, 338-353.
117. Zadeh, L.A. (1978). Fuzzy Sets as a Basis for a Theory of Possibility. Fuzzy Sets and Systems 1:3–28.