



TECHNICAL REPORT

CTR/34/04

October 2004

Genetic algorithm based technique for solving chance constrained problems arising in risk management.

Boby Varghese and Chandra Poojari

GENETIC ALGORITHM BASED TECHNIQUE FOR SOLVING CHANCE CONSTRAINED PROBLEMS ARISING IN RISK MANAGEMENT [★]

Boby Varghese and Chandra A. Poojari ^{*}

*Centre for the analysis of Risk and optimisation modelling applications
(CARISMA), Department of Mathematical Sciences, Brunel University*

Abstract

Management and measurement of Risk is an important issue in almost all areas involving decision making under uncertainty. Stochastic Programming (SP) techniques is used to model the uncertainties and construct the mathematical model in number of application domains. We discuss alternate SP modelling paradigm such as recourse and chance constraint programming models. In particular we investigate alternate formulations of chance constraint programming models. We investigate and develop a chance constraint formulation for the pension fund models, and carry out detailed computational investigation. We develop and implement a Genetic Algorithm and Monte-Carlo sampling based framework for chance constrained programming problems. We have bench-marked the framework on alternate deterministic non-linear and chance constrained programming models. We found that the framework can successfully process highly nonlinear, non-convex deterministic and stochastic optimization problems. The disadvantage of using genetic algorithm is that optimal solution is not guaranteed within a finite computing time.

1 Background

Many decision problems can be modelled using mathematical programs, which seek to maximize or minimize an objective which is a function of the decisions. The possible decisions are constrained by limits in resources (labour, budget)

^{*} We are grateful to Brunel University's BRIEF award. The grant allowed Mr. Bobby Varghese to pursue his M.Phil and carry out the research investigation discussed in this report.

^{*} Chandra.Poojari@brunel.ac.uk

and minimum requirements. Stochastic programs are mathematical programs where some of the data incorporated into the objective or constraints are uncertain. Uncertainty is usually characterized by a probability distribution on the parameters. Although the uncertainty is rigorously defined, in practice it can range in detail from a few scenarios (possible outcomes of the data) to specific and precise joint probability distributions. Stochastic optimization problems occur in many applied problems of engineering design, statistics, electric power generation and planning, pollution redevication, structural engineering, industrial capacity planning, financial risk management and investment, business management etc. Stochastic procedures for solving problems of this kind are often considered and several concepts are proposed to ensure and improve the convergence behavior of developed methods (Shapiro [45,46], Kall and Wallace [20], Prekopa [37]).

Stochastic programming models can include both *anticipative* and *adaptive* decision variables. Anticipative variables corresponds to those decisions that must be made *here-and-now* and cannot depend on the future observations/partial realizations of the random parameters. Adaptive variables corresponds to *wait-and-see* decisions after some/all of the random parameters are observed. A generic anticipative stochastic programming problem can be formulated as follows:

$$\min: E [f(\mathbf{x}, \xi)] \quad (1)$$

subject to

$$E[g_j(\mathbf{x}, \xi)] \leq 0, j = 1, 2, \dots, k \quad (2)$$

$$E[h_j(\mathbf{x}, \xi)] = 0, j = 1, 2, \dots, r \quad (3)$$

$$\mathbf{x} \in \mathbf{X} \quad (4)$$

In an adaptive model, information related to uncertainty becomes partially available before decision making, so optimization takes place in a learning environment, which is the essential difference with an anticipative model. An adaptive stochastic program can be formulated as:

$$\min: E [f(\mathbf{x}, \xi)|\mathcal{A}] \quad (5)$$

subject to

$$E [g_j(\mathbf{x}, \xi)|\mathcal{A}] \leq 0, j = 1, 2, \dots, k \quad (6)$$

$$E [h_j(\mathbf{x}, \xi)|\mathcal{A}] = 0, j = 1, 2, \dots, r \quad (7)$$

$$\mathbf{x} \in \mathbf{X} \quad (8)$$

E denote the expectation operator.

$\mathbf{x} = (x_1, x_2, \dots, x_n)$ is an n -dimensional vector defined over the set \mathbb{R}^n , $\xi = (\xi_1, \xi_2, \dots, \xi_m)$ is a m -dimensional continuous stochastic vector defined over the set \mathbb{R}^m , and $f, g_j, j = 1, 2, \dots, k$ and $h_j, j = 1, 2, \dots, r$, are real valued functions defined on \mathbb{R}^{n+m} . \mathbf{X} represents (implicitly) the constraints that do not depend on the random parameters ξ .

In anticipative model (also referred to as static model), when the constraints

depend on random parameters, feasibility is often expressed in terms of *probabilistic or chance constraints*. A typical formulation would require that a particular variable/function lie within a target range with a certain probability or that the expected value of a penalty for not meeting the target be combined with the objective and optimized (this is called a problem with *recourse*). For technical reasons (chance constrained problems often lead to nonlinear, even nonconvex, constraints) by far the greater number of applications are formulated as recourse problems.

We consider two classes of stochastic programs: *recourse programs* and *chance constrained programs*.

Most research in stochastic programming, has been devoted to linear, convex optimization problems and the solution technique heavily rely on the scenario approach. With advance of computing power, computationally intensive algorithms have been applied in the field of stochastic programming. The use of evolutionary algorithms into this area has been limited. In this paper, we investigate the use of genetic algorithms (GAs) for decision making under uncertainty, in particular, we develop a GA and Monte Carlo Sampling for chance constrained programming.

2 Chance constrained Programming

(CCP) Chance constrained programming (CCP) was pioneered by Charnes and Cooper [1]. They suggested three models that have different objective functions and probabilistic type of constraints, a model that maximizes the expected value of the objective function (the **E**-model), a model that minimizes the generalized mean square of the objective function (the **V**-model), and a model that maximizes the probability of exceeding an aspiration level of the objective function (the **P**-model). CCP is an anticipative stochastic optimization approach, which can deal explicitly with variability. In CCP, the focus is on the reliability of the system, *i.e.*, the system's ability to meet the constraints (risk measures) with certain reliability in an uncertain environment. Through the use of standard probability distribution models, which are analytically tractable, it is possible to convert the chance constraints into deterministic equivalents. This enables CCP to be implemented using standard mathematical programming packages. CCP techniques have been applied to many financial [16,53], environmental [21], transportation [44], construction, energy, chemical processing [35,15] aerospace, and military system [48] optimization problems.

2.1 Alternate CCP Formulation

Many optimization problems arising in risk management can be modelled using CCP. For the convenience of optimization, the objective function is assumed in the form given below:

$$\min: E[f(\mathbf{x}, \xi)] \quad (9)$$

or

$$\max: P[f(\mathbf{x}, \xi) \leq u] \quad (10)$$

where u is a specified risk level (probability maximization)
subject to

$$E[g_j(\mathbf{x}, \xi)] \leq 0, j = 1, 2, \dots, k \quad (11)$$

$$E[h_j(\mathbf{x}, \xi)] = 0, j = 1, 2, \dots, r \quad (12)$$

(expectation constraint form)

$$P[g_j(\mathbf{x}, \xi) \leq 0] \geq \alpha_j, j = 1, 2, \dots, k \quad (13)$$

$$P[h_j(\mathbf{x}, \xi) = 0] \geq \beta_j, j = 1, 2, \dots, r \quad (14)$$

α_j, β_j are the minimum probability measure (reliability/safety index) of the events $g_j(\mathbf{x}, \xi) \leq 0, h_j(\mathbf{x}, \xi) = 0$ respectively (individual/separate probability constraint (IPC/SPC) form)

$$P[g_1(\mathbf{x}, \xi) \leq 0, g_2(\mathbf{x}, \xi) \leq 0, \dots, g_k(\mathbf{x}, \xi) \leq 0] \geq \alpha \quad (15)$$

$$P[h_1(\mathbf{x}, \xi) = 0, h_2(\mathbf{x}, \xi) = 0, \dots, h_r(\mathbf{x}, \xi) = 0] \geq \beta \quad (16)$$

(joint probability constraint(JPC) form)

E denotes the expectation operator.

$\mathbf{x} = (x_1, x_2, \dots, x_n)$ is an n -dimensional vector defined over the set \mathbb{R}^n ,
 $\xi = (\xi_1, \xi_2, \dots, \xi_m)$ is a m -dimensional continuous stochastic vector defined over the set \mathbb{R}^m , and $f, g_j, j = 1, 2, \dots, k$ and $h_j, j = 1, 2, \dots, r$, are real valued functions defined on \mathbb{R}^{n+m} .

Calculation of the expected values and probabilities can be carried out by applying two methods: either the first order reliability method (FORM) or Monte Carlo Simulation. As Monte Carlo is a time-consuming process, the FORM is usually used to compute the probability.

2.2 Transformation of Chance Constraints

When the parameters in a linear CCP problem are normally or exponentially distributed with known means and variances, the stochastic problem can be transformed into an equivalent deterministic nonlinear programming

problem. The deterministic equivalents are obtained for both individually [50] and jointly [43] constrained linear CCP problems as given below.

Linear CCP problem with individual chance constraints A CCP with individual probability constraint can be stated as

$$Max : z(\mathbf{x}) = \sum_{j=1}^n c_j x_j \quad (17)$$

subject to

$$P \left[\sum_{j=1}^n a_{ij} x_j \leq b_i \right] \geq 1 - \alpha_i \quad (18)$$

$$x_j \geq 0, j = 1, 2, \dots, n$$

$$\alpha \in (0, 1), i = 1, 2, \dots, m$$

If a_{ij} , b_i , c_j are normally distributed random variables with known mean and variances. There are seven subcases arising which are as follows:

- (1) only a_{ij} are random variables
- (2) only b_i are random variables
- (3) only c_j are random variables
- (4) a_{ij} , b_i random variables
- (5) a_{ij} and c_j are random variables
- (6) b_i and c_j are random variables
- (7) a_{ij} , b_i and c_j are random variables

Case I: only a_{ij} are random variables

Let $E(a_{ij})$ and $Var(a_{ij})$ be the mean and the variances of the normal random variable a_{ij} assume that covariance between a_{ij} and a_{ik} is known

$$d_i = \sum_{j=1}^n a_{ij} x_j, i = 1, 2, \dots, m$$

d_i is normal rv with

$$E(d_i) = \sum_{j=1}^n E(a_{ij}) x_j, i = 1, 2, \dots, m$$

and

$$Var(d_i) = X^T V_i X, i = 1, 2, \dots, m$$

where V_i is the i^{th} covariance matrix defined as

$$V_i = \begin{pmatrix} Var(a_{i1}) & Cov(a_{i1}, a_{i2}) & \dots & Cov(a_{i1}, a_{in}) \\ Cov(a_{i2}, a_{i1}) & Var(a_{i2}) & \dots & Cov(a_{i2}, a_{in}) \\ \dots & \dots & \dots & \dots \\ Cov(a_{in}, a_{i1}) & Cov(a_{in}, a_{i2}) & \dots & Var(a_{in}) \end{pmatrix} \quad (19)$$

above constraint can be expressed as

$$P[d_i \leq b_i] \geq 1 - \alpha_i, i = 1, 2, \dots, m \quad (20)$$

above equation can be written as

$$P[d_i \leq b_i] = \Phi \left[\frac{b_i - E(d_i)}{\sqrt{var(d_i)}} \right], i = 1, 2, \dots, m \quad (21)$$

where $\Phi(z)$ represents the cdf of the standard normal random variable evaluated at z . If

$$\Phi(K_{\alpha_i}) = 1 - \alpha_i$$

then 2.3 can be written as

$$\Phi \left[\frac{b_i - E(d_i)}{\sqrt{var(d_i)}} \right] \geq \Phi(K_{\alpha_i}), i = 1, 2, \dots, m \quad (22)$$

$$\frac{b_i - E(d_i)}{\sqrt{var(d_i)}} \geq K_{\alpha_i}, i = 1, 2, \dots, m \quad (23)$$

or

$$E(d_i) + K_{\alpha_i} \sqrt{Var(d_i)} \leq b_i, i = 1, 2, \dots, m$$

Thus the probabilistic LP is equivalent to solving the deterministic NLP problem:

$$Max : z(\mathbf{x}) = \sum_{j=1}^n c_j x_j \quad (24)$$

subject to

$$\sum_{j=1}^n E(a_{ij})x_j + K_{\alpha_i} \sqrt{\mathbf{x}^T V_i \mathbf{x}} \leq b_i, i = 1, 2, \dots, m \quad (25)$$

if a_{ij} are independent covariance terms will be zero and above constraint can be simplified as

$$\sum_{j=1}^n E(a_{ij})x_j + K_{\alpha_i} \sqrt{\sum_{j=1}^n Var(a_{ij})x_j^2} \leq b_i, i = 1, 2, \dots, m \quad (26)$$

$$x_j \geq 0, j = 1, 2, \dots, n$$

Case II: only b_i are random variables

The constraint 18 can be restated as

$$P \left[\frac{b_i - E(b_i)}{\sigma(b_i)} \leq \frac{\sum_{j=1}^n a_{ij}x_j - E(b_i)}{\sigma(b_i)} \right] \leq \alpha_i, \quad (27)$$

or

$$\Phi \left[\frac{b_i - E(b_i)}{\sigma(b_i)} \right] \leq \Phi [K_{1-\alpha_i}]$$

This inequality will be satisfied only if

$$\left[\frac{b_i - E(b_i)}{\sigma(b_i)} \right] \leq [K_{1-\alpha_i}]$$

or

$$\sum_{j=1}^n a_{ij}x_j \leq E(b_i)K_{1-\alpha_i}\sqrt{Var(b_i)}, i = 1, 2, \dots, m \quad (28)$$

Thus the probabilistic LP is equivalent to the following deterministic LP problem

$$Max : z(\mathbf{x}) = \sum_{j=1}^n c_jx_j \quad (29)$$

subject to

$$\begin{aligned} \sum_{j=1}^n a_{ij}x_j &\leq E(b_i)K_{1-\alpha_i}\sqrt{Var(b_i)}, i = 1, 2, \dots, m \\ x_j &\geq 0, j = 1, 2, \dots, n \end{aligned} \quad (30)$$

Case III: Only c_j are random variables

Since c_j are normally distributed random variables, the objective function $z(\mathbf{x})$, will also be a normally distributed random variable. The mean of $z(\mathbf{x})$ is given by

$$E[z(\mathbf{x})] = \sum_{j=1}^n E(c_j)x_j \quad (31)$$

where $E(c_j)$ is the mean value of c_j . Thus the deterministic problem can be formulated as an **E**-model:

$$Max : E[z(\mathbf{x})] = \sum_{j=1}^n E(c_j)x_j \quad (32)$$

subject to

$$\begin{aligned} \sum_{j=1}^n a_{ij}x_j &\leq b_i, i = 1, 2, \dots, m \\ x_j &\geq 0, j = 1, 2, \dots, n \end{aligned}$$

Case IV: a_{ij} , b_i and c_j are random variables

The constraint 18 can be restated as

$$P \left[\frac{h_i - E(h_i)}{\sigma(h_i)} \leq \frac{-E(h_i)}{\sigma(h_i)} \right] \geq 1 - \alpha_i, i = 1, 2, \dots, m \quad (33)$$

where h_i is normal random variable defined as

$$h_i = \sum_{j=1}^n a_{ij}x_j - b_i, i = 1, 2, \dots, m$$

the above inequality can be written as

$$\Phi \left[\frac{-E(h_i)}{\sigma(h_i)} \right] \geq \Phi [K_{\alpha_i}], i = 1, 2, \dots, m$$

the above inequality will be satisfied only if

$$\frac{-E(h_i)}{\sigma(h_i)} \geq K_{\alpha_i}, i = 1, 2, \dots, m$$

$$E(h_i) + K_{\alpha_i} \sqrt{Var(h_j)} \leq 0, i = 1, 2, \dots, m$$

Thus the probabilistic LP is equivalent to the following deterministic NLP problem

$$Max : z(\mathbf{x}) = \sum_{j=1}^n c_j x_j \quad (34)$$

subject to

$$E(h_i) + K_{\alpha_i} \sqrt{Var(h_j)} \leq 0, i = 1, 2, \dots, m \quad (35)$$

$$x_j \geq 0, j = 1, 2, \dots, n$$

The other cases can be obtained by the combinations of these four cases.

Linear CCP problem with a joint chance constraint A CCP with joint probability constraint can be stated as

$$Max : z(\mathbf{x}) = \sum_{j=1}^n c_j x_j \quad (36)$$

$$P \left[\bigcap_{i=1}^m \left\{ \sum_{j=1}^n a_{ij} x_j \leq b_i \right\} \right] \geq 1 - \alpha, \quad (37)$$

joint probabilistic constraint

$$x_j \geq 0, j = 1, 2, \dots, n$$

Where b_i 's are independent normal random variables with known means and variances and $0 \leq \alpha \leq 1$ with a specified probability. The above constraint can be written as

$$\prod_{i=1}^m Pr \left[\sum_{j=1}^n a_{ij} x_j \leq b_i \right] \geq 1 - \alpha, \quad (38)$$

$$\prod_{i=1}^m Pr \left[\frac{b_i - E(b_i)}{\sigma(b_i)} \leq \frac{\sum_{j=1}^n a_{ij} x_j - E(b_i)}{\sigma(b_i)} \right] \geq 1 - \alpha, \quad (39)$$

where $\frac{b_i - E(b_i)}{\sigma(b_i)}, \forall i = 1, \dots, m$ is a standard normal variate with zero mean and unit variance. Hence,

$$\prod_{i=1}^m \Phi \left[\frac{\sum_{j=1}^n a_{ij} x_j - E(b_i)}{\sigma(b_i)} \right] \geq 1 - \alpha, \quad (40)$$

where Φ represents the c.d.f of standard normal variable. The equivalent deterministic model of the probabilistic LP problem can be presented as the deterministic NLP problem:

$$Max : z(\mathbf{x}) = \sum_{j=1}^n c_j x_j \quad (41)$$

subject to

$$\sum_{j=1}^n a_{ij} x_j - \beta_i \sigma(b_i), i = 1, \dots, m \quad (42)$$

$$\frac{3\beta_i}{3 - \beta_i^2} \exp(-\beta_i^2/2) \geq \sqrt{\frac{\pi}{2}} (2y_i + 1), i = 1, \dots, m \quad (43)$$

$$\prod_{i=1}^m y_i \geq 1 - \alpha \quad (44)$$

$$0 \leq y_i \leq 1, i = 1, \dots, m \quad (45)$$

$$y_j \geq 0, j = 1, \dots, n \quad (46)$$

Thus a probabilistic LP problem with joint probability constraints can be easily transformed into a deterministic NLP problem. For proof please refer to [43].

Linear CCP problem with exponential random variables When the a_{ij} and c_j are independent exponential random variables with known distributions, the linear CCP problem can be transformed into a deterministic nonlinear programming problem. Let $E(a_{ij}) = \frac{1}{\lambda_{ij}}, \lambda_{ij} > 0, j = 1, 2, \dots, n$. The probability density function of the random variable $Y_i = \sum_{j=1}^n a_{ij} x_j$ is given by

$$g_i(y_i) = \prod_{j=1}^n \lambda_{ij} \left[\sum_{k=1}^n \frac{x_k^{n-2} e^{-\lambda_{ik} y_i / x_k}}{\prod_{l=1, l \neq k}^n (x_k \lambda_{il} - x_l \lambda_{ik})} \right] \quad \text{if } y_i > 0 \quad (47)$$

A proof by induction [26] is given in the Appendix. The deterministic constraint can be obtained by integrating the pdf of Y_i as stated below.

$$\int_0^{b_i} g_i(y) dy_i \geq 1 - \alpha_i, \quad i = 1, \dots, m. \quad (48)$$

The above integral can be simplified as follows:

$$\prod_{j=1}^n \lambda_{ij} \left[\sum_{k=1}^n \frac{x_k^{n-1} e^{-\lambda_{ik} y_i / x_k}}{\lambda_{ik} \prod_{l=1, l \neq k}^n (x_k \lambda_{il} - x_l \lambda_{ik})} \right] \leq \alpha_i, \quad i = 1, \dots, m \quad (49)$$

Thus the deterministic model of the linear CCP is

$$Max : z(\mathbf{x}) = \sum_{j=1}^n E [c_j] x_j \quad (50)$$

subject to

$$\prod_{j=1}^n \lambda_{ij} \left[\sum_{k=1}^n \frac{x_k^{n-1} e^{-\lambda_{ik} y_i / x_k}}{\lambda_{ik} \prod_{l=1, l \neq k}^n (x_k \lambda_{il} - x_l \lambda_{ik})} \right] \leq \alpha_i, \quad i = 1, \dots, m \quad (51)$$

$$x_j \geq 0, j = 1, 2, \dots, n$$

2.3 Stochastic Metrics

For stochastic optimization problems, we compute two statistics which quantify the importance of randomness. The value of a stochastic solution (VSS) is the difference in the objective function for the stochastic problem (call it the stochastic solution, SS or here-and-now, HN) and the objective value for the deterministic problem computed by replacing stochastic variables by their expectations (call it the expected value solution, EVS): ie, if

$$SS = E_{\xi}[f(x^*, \xi)],$$

$$x^* \text{ optimal solution of } \min_{x \in X} E_{\xi}[f(x, \xi)]$$

$$EVS = E_{\xi}[f(x^*(E\xi), \xi)] = f(x^*(E\xi), E\xi),$$

$$x^*(E\xi) \text{ optimal solution of } \min_{x \in X} f(x, E\xi) \text{ then,}$$

$$VSS = SS - EVS$$

This computes the benefits of knowing the stochastic variables. We also compute the expected value with perfect information or wait-and-see solution (WSS). The difference between WSS and VSS is the expected value of perfect information (EVPI): ie, if

$$WSS = E_{\xi}[f(x^*(\xi), \xi)],$$

$$x^*(\xi) \text{ optimal solution of } \min_{x \in X} f(x, \xi) \text{ then,}$$

$$EVPI = WSS - SS$$

2.4 Algorithms and Computational Challenges

The main challenges in designing algorithms for stochastic programming arises from the need to calculate conditional expectation and/or probability associated with multi-dimensional random variables. The computational challenges associated with SP vary a great deal with the class of problems being discussed. As with any large scale optimization problem, exploiting properties and structure of the problems provides the key to effective algorithms. Even in cases where the random variables are discrete, the total number of outcomes of a multi-dimensional random vector can be so large that the calculations associated with the summations can be far too demanding. Hence even in the case of discrete random variables we have to resort to approximations. Discretization and/or aggregations in multi-stage problems [33,13,30] result in alternative sample paths or data scenarios that allow us to control the growth of the scenario tree.

A general CCP problem involves an objective function with several linear/nonlinear joint/individual probabilistic constraints. The resulting problem can be classified as a nonlinear programming problem where the nonlinear constraint involves a multidimensional integral. The computation of the probabilistic constraint and its gradient involves Monte Carlo integration, which results in inaccurate function values and gradients. Non convexity of objective function and constraints is another issue. Another source of severe numerical difficulties has its roots in the nature of probability distribution functions: the magnitude of the components of the gradient is usually very small outside a narrow region. The algorithms for the solution of probabilistically constrained problems are based on general nonlinear programming algorithmic schemes, which are specialized to utilize the structure of problems with joint/individual probabilistic constraints. For certain distributions (including multinormal) and certain probability levels the linear CCP problem with separate chance constraints turns out to be a convex programming problem. In the general case for certain multivariate distributions and probability levels convex programming algebraic equivalents exist. Prekopa [36] discusses several approaches for solving chance-constrained models including gradient methods and the use of penalty functions.

Another approach is to use an evolution program for solving the CCP problem. Genetic Algorithms can be used to solve extremely complex search and optimization problems that are difficult to handle using analytic or simple enumeration methods. The design of a simulation based GA is given chapter 5.

3 Risk Management

Risk Management is the act of taking effective and reasonable steps to minimize risk. It is a systematic approach to setting the best course of action under uncertainty by identifying, assessing, understanding, and acting on and communicating risk issues. The modelling of risk is at present the most challenging risk management tasks in quantitative finance. There are various measures used to evaluate risk in finance (especially in portfolio optimization). Each give the portfolio manager a different level of information. The ultimate objective of every manager is clearly to obtain a profit as a result of risk management.

3.1 Definition and Classification of Risk

Risk refers to the loss due to uncertainty that surrounds future events and outcomes. It is the expression of the likelihood and impact of an event with the potential to influence the achievement of an organization's objectives. Risk can be defined as a trinity of risk event ω , risk probability P , function of risk losses u :

$$R = (\omega, P, u)$$

A risk event ω is a random event that is connected with any project decision, risk probability P is the probability of the risk event, loss function u is the function $u(\omega)$ that defines the damage if the risk event $\omega \in \Omega$, where Ω is a set of risk events, occurs. In the continuous case, risk is defined as the mathematical expectation of damage

$$Risk = \int_{\Omega} u(\omega)p(\omega)d\omega$$

In the discrete case, risk is defined as

$$Risk = \sum_{i=1}^n u_i p_i$$

where p_i is the probability that the hazard event i will occur and u_i is the damage and the cardinality of set Ω , $|\Omega| = n$. The major types of risk are - (1) Objective risk, (2) Subjective risk, (3) Perceived Risk.

Objective risk, R_o , and objective probability, p_o , which is the property of real physical systems. Subjective risk, R_s , and subjective probability, p_s , which is the degree of belief in a statement. R_s and p_s are not properties of the physical systems under consideration (but may be some function of R_o and p_o). Perceived Risk, R_p , which is related to an individual's feeling of fear in the face of an undesirable possible event, is not properties of the physical systems

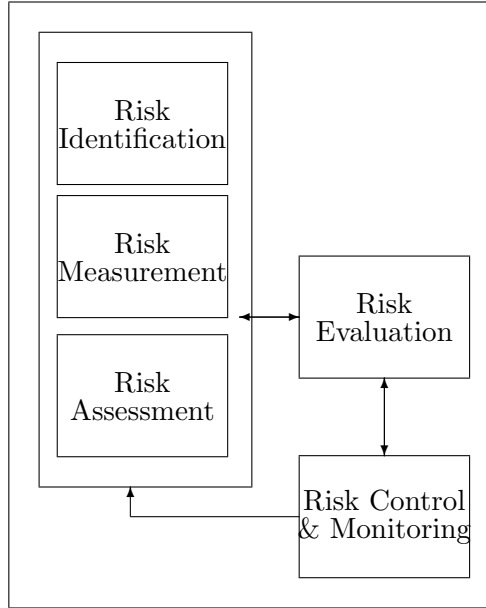


Fig. 1. Risk Management Framework

but is related to fear of the unknown. It may be a function of R_o , p_o , R_s , and p_s .

3.2 Risk Management Framework

Risk management is the focused management of risks to the achievement of an organizations operational objectives, and involves the following six activities, classified under two categories 1) risk assessment, 2) risk control. A schematic representation of the risk management framework is given in figure 1.

Risk Assessment

- i Risk identification: this involves tracking of the risk through data compilation and analysis. Risk Identification methods should have the following attributes:- (1) they should examine all areas of the system in a systematic manner, (2) they should be proactive rather than reactive, and (3) they should synthesize risk information from all available sources of risk information.
- ii Risk analysis: This involves defining the risk event, calculating the probability of the risk event, and then estimating the mean damage of the risk event.
- iii Risk prioritization: The objective of Risk Prioritization is to prioritize the identified risks for mitigation. Both qualitative and quantitative methods can be used to categorize the risks as to their relative severity and po-

tential impact on the project. To effectively compare identified risks, and to provide a proactive perspective, the risk prioritization method should consider the following factors:- (1) the probability of the risk occurring, (2) the consequence of the risk, and (3) the cost and resources required to mitigate the risk. At the conclusion of risk prioritization, a consolidated list of risks is created.

Risk Control

- i Risk management planning: This involves the following steps:- (1) prototype risk management plan, (2) assigning responsibility and accountability, and (3) choosing the right tools and techniques.
- ii Risk resolution: Once the probability of system success/failure or the loss due to the uncertainty is calculated, the model can be used to study alternatives that decreases the probability of failure or the loss associated with it. This step, or the whole process can be iterated until a solution is achieved.
- iii Risk monitoring: this is to control and monitor the risk, communicate and document the process and decisions.

Effective risk management implementation is continuous and iterative, requiring an on-going process of risk identification, prioritization, resolution and monitoring.

3.3 Mathematical Techniques for Risk Analysis and Quantification

Practical measurements of risk are extremely important for risk management. Theories and algorithms of simulation, optimization, multi-objective analysis and decision making under uncertainty are all applicable to risk management including both, stochastic and fuzzy set approaches. The following is a presentation of available techniques for risk management [49].

Stochastic Simulation Simulation models describe the state of the system for various inputs. They, however, give no direct measure of what decision would be taken to improve the performance of the system. Major components of a simulation model are:

- Input - quantities that drive the model
- Physical relationship-mathematical expression of the relationship among the physical variables of the system being modelled
- Non-physical variables-those that defines economic variables etc.
- Operation rules - rules that govern operational control
- Outputs-the final product of operations on inputs by the physical and non-physical relations in accordance with operating rules.

Stochastic Optimization Generally there are two categories of stochastic optimization techniques:

- (1) Implicit - which use deterministic models with the generated sequence of random variables/vectors called scenarios. This approach is usually used for solving multistage stochastic recourse problems.
- (2) Explicit - which incorporate uncertainty directly in the objective function and/or constraints. In this case, the joint/individual probability distribution function of the random variables will be known.

Stochastic Multi-Objective Analysis Many practical instances of decision making involve replacing single-objective optimization with multi-objective analysis. The importance of considering multiple objective at a time is emphasized by the existence of several conflicting and incommensurable objectives in every step of the risk management process. A multi-objective programming problem may be given as:

$$\begin{array}{l} \text{Min :} \\ \mathbf{x} \in \mathbf{X} \end{array} \mathbf{Z}(\mathbf{x}, \xi) = [Z_1(\mathbf{x}, \xi), Z_2(\mathbf{x}, \xi), \dots, Z_p(\mathbf{x}, \xi)]$$

where \mathbf{X} represents the feasible solution set determined by the problem constraints,

$\mathbf{Z} = (Z_1, Z_2, \dots, Z_p)$ is a p -dimensional vector of objectives,

$\mathbf{x} = (x_1, x_2, \dots, x_n)$ is an n -dimensional vector defined over the set \mathbb{R}^n ,

$\xi = (\xi_1, \xi_2, \dots, \xi_m)$ is a m -dimensional continuous stochastic vector defined over the set \mathbb{R}^m .

Above formulation of multi-objective problem may include uncertainties of different nature. It has been shown that it is possible to deal with random variables in constraints and objective functions of stochastic multi-objective programming problem. The attitude of the decision maker towards risk can be directly incorporated into the analysis through the consideration of risk as one of the objective functions $Z(\mathbf{x}, \xi)$.

Other tools used for risk management are stochastic modelling, fuzzy optimization, decision theory, regression analysis, fuzzy multi-objective analysis, neural networks, rule-based expert systems, genetic algorithms, simulated annealing, dynamic search space reduction.

3.4 Measures of Risk

Measuring risk is at the core of modern risk management. We give an overview of the existing approaches to risk quantification. Mathematically, a risk measure ρ is a function mapping distribution of losses \mathcal{G} to \mathbb{R} (i.e. $\rho : \mathcal{G} \rightarrow \mathbb{R}$). Various risk measures and their properties are given by Albrecht[2] and Urya-

sev et. al [52,41,11].

3.5 Risk as Magnitude of the Distance from a Target

Let a financial consequence u such as loss, reward, demand etc be represented as a stochastic function $f(x, \xi)$ where x is decision vector (eg. number of units bought, fraction of asset invested) and ξ is a vector of uncertain variables (eg. random returns, uncertain prices).

Two-Sided risk Measures Two-sided risk measures measure the magnitude of the distance (in both directions) from the realizations of u to $E(u)$. Different functions of distance lead to different risk measures. Looking at quadratic deviations (volatility) leads to the risk measure variance given by

$$Var(u) = E [(u - E(u))^2]$$

Two sided risk measures contradicts the intuitive notion of risk that only negative deviations are dangerous, only downside risk that matters. In addition variance does not account for the fat tails of the underlying distribution and for the corresponding tail risk. This leads to higher (normalized) central moments (eg. skewness and kurtosis) being included in the analysis to asses risk more properly. Considering the absolute deviation as a measure of distance, we obtain the mean absolute deviation(MAD)- measure

$$R(u) = E [|u - E(u)|]$$

or more general risk measures of degree k like,

$$R(u) = E [|u - E(u)|^k]$$

and

$$R(u) = E [|u - E(u)|^k]^{1/k}$$

Measures of Shortfall Risk Measures of shortfall risk are one-sided risk measures and measure the shortfall risk (downside risk) relative to the target variable. This target may be the expected value, but in general it is an arbitrary deterministic target z (target gain, target return, minimum acceptable return) or even a stochastic benchmark. A general class of risk measures is the class of lower partial moments of degree k .

$$LPM_k(z, u) = E [max(z - u, 0)^k]$$

or, in normalized form($k \geq 2$)

$$R(u) = LPM_k(z, u)^{1/k}$$

Basic cases playing important role in applications, are obtained for $k = 0, 1, 2$. These are *shortfall probability*

$$SP_z(u) = P(u \leq z) = F(z)$$

the *shortfall expectation*

$$SE_z(u) = E [\max(z - u, 0)]$$

and the *shortfall variance*

$$SV_z(u) = E [\max(z - u, 0)^2]$$

Variations are obtained for $z = E(X)$ *lower-semi-absolute-deviation(LSAD)*

$$R(u) = E [\max(E(u) - u, 0)]$$

the *semivariance*

$$R(u) = E [\max(E(u) - u, 0)^2]$$

Another variation is to consider conditional measures of shortfall risk. The *mean excess loss*(conditional shortfall expectation)

$$MEL_z(u) = E(z - u | u \leq z) = \frac{SE_z(u)}{SP_z(u)}$$

the average shortfall under the condition that a shortfall occurs.

Despite the advantage of being closer to an intuitive notion of risk, shortfall measures have the disadvantage that they lead to greater technical problems with respect to the disaggregation of portfolio risk, optimization and statistical identification.

3.6 Risk as Necessary Capital or Necessary Premium

Value at Risk (VaR) The VaR at confidence level α and time horizon t , given the probability distribution of the future loss of a financial institution, is the quantity such that in less than $(1 - \alpha)\%$ of the cases the loss will exceed this level. In practice, the time horizon used is ten day period and the α -level is quite high, typically 99% or 95%. A rigorous definition of VaR is given below: Let $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ be a loss function which depends on the decision vector x belonging to the feasible set $X \subset \mathbb{R}^n$ and a random vector $\xi \in \mathbb{R}^m$. The vector x can be interpreted as representing a portfolio, with X as the set of available portfolios (subject to various constraints). Suppose the random vector ξ is governed by a probability measure P that is independent of x . Consider that the random vector ξ has the probability distribution function

$p : \mathbb{R}^m \rightarrow \mathbb{R}$. For each x , denote by $\Phi(x, \cdot)$ on \mathbb{R} the resulting distribution function for the loss $z = f(x, \xi)$, i.e.,

$$\Phi(x, VaR) = P \{ \xi | f(x, \xi) \leq VaR \} \quad (52)$$

$$= \int_{f(x, \xi) \leq VaR} p(\xi) d\xi \quad (53)$$

By definition, $\Phi(x, VaR)$ is the probability that the loss function $f(x, \xi)$ does not exceed the threshold VaR . The VaR function, $VaR_\alpha(x)$, which is a quantile of the loss distribution is defined in the following way

$$VaR_\alpha(x) = \min \{ VaR \in \mathbb{R} : \Phi(x, VaR) \geq \alpha \} \quad (54)$$

There are many methods to calculate VaR, which fit different market conditions, data set and precision requirements. Generally we can classify them into three types:

- variance-covariance method
- historical simulation method
- Monte Carlo simulation method

The detailed discussion about historical and Monte Carlo simulation methods was given by Dowd [10]. Monte Carlo simulation is frequently used to estimate the VaR. One greatest drawback with Monte Carlo simulation is its large requirement of computer resources.

Conditional Value-at-Risk (CVaR) For continuous distributions, CVaR is defined as the conditional expected value of losses under condition that they exceed VaR, which is denoted by $CVaR_\alpha(x)$. We can write the CVaR function, $CVaR_\alpha(x)$, as follows

$$CVaR_\alpha(x) = E [f(x, \xi) | f(x, \xi) \geq VaR_\alpha(x)]$$

When the loss distribution has no jumps, CVaR function can be written as

$$CVaR_\alpha(x) = (1 - \alpha)^{-1} \int_{f(x, \xi) \geq VaR_\alpha(x)} f(x, \xi) p(\xi) d\xi \quad (55)$$

For general distributions, including discrete distributions, CVaR is defined as the weighted average of VaR and losses strictly exceeding VaR. Denote by $CVaR_\alpha(x)^+$ the conditional expectation of losses strictly exceeding VaR. For general distributions, consider the augmented function,

$$F_\alpha(x, VaR) = \lambda VaR_\alpha(x) + (1 - \lambda) CVaR_\alpha(x)^+ \quad (56)$$

where $\lambda = [\Phi(x, VaR_\alpha(x)) - \alpha] / [1 - \alpha] \in [0, 1]$ [47]. Under the assumption that the loss function $f(x, \xi)$ is convex, and it can be shown that $F_\alpha(x, VaR)$

is convex and continuously differentiable with respect to VaR and $CVaR_\alpha(x)$ is convex with respect to x . Moreover, minimizing the CVaR over any $x \in X$, is equivalent to minimizing $F_\alpha(x, VaR)$ over $(x, VaR) \in X \times \mathbb{R}$, i.e.,

$$\min_{x \in X} CVaR_\alpha(x) \equiv \min_{(x, VaR) \in X \times \mathbb{R}} F_\alpha(x, VaR) \quad (57)$$

If, in addition, X is a convex set, then the CVaR minimization problem

$$\min_{(x, VaR) \in X \times \mathbb{R}} F_\alpha(x, VaR) \quad (58)$$

is a convex programming problem. CVaR is a coherent percentile risk measure and is the preferred risk measure to VaR. CVaR is a sub-additive measure of risk compared with VaR which is not sub-additive. Sub-additivity means diversification of portfolio reduces CVaR, but may increase VaR. Also, VaR does not provide any information about the amount of loss exceeding VaR. The reader is referred to [2] and Uryasev et. al [52,41,11] for proof.

4 A Genetic Algorithm Based Computational Framework for Stochastic Programming

In this chapter a method for solving stochastic optimization problems by a Genetic Algorithm(GA) and finite series of Monte-Carlo samples is considered. We use a multiplicative penalty function to penalize the infeasibility of the solutions. The design of the genetic algorithm is investigated in detail. Several numerical experiments are carried out to show the effectiveness of the algorithm for nonlinear stochastic programming.

4.1 Overview of Genetic Algorithms

GAs have been applied to optimization problems in many fields, such as optimal control problems, job scheduling, transportation problems, pattern recognition, machine learning and so on [29,12,27,24]. It is a powerful search algorithm, which starts with an initial set of random potential solutions called the population and uses a process similar to biological evolution to improve upon them. These are robust algorithms that are capable of optimizing multi-modal, noisy, dynamic functions. They can be distinguished from similar search and optimization procedures by the following characteristics:

- They operate on a representation of the parameters to be modified and not on the parameters directly.

- GAs search in a parallel manner with a multi-individual population being considered at the same time instead of the individuals being examined one by one. Hence they work well in parallel implementation.
- GAs do not require specific information about the function (such as first or second derivatives) and only need some way of evaluating the fitness of solutions (payoff information).
- The solutions are modified in an iterative manner using probabilistic operators, not deterministic ones. GAs constitute a directed global random search procedure.
- They work with various data types.
- They create a list of potential optimal solutions, not just a single solution.

A simple GA works as follows: GAs start out with an initial *population* of possible solutions (individuals) to a given problem (the environment) where each individual is represented using some form of encoding as a *chromosome*. These chromosomes are evaluated in some way for their *fitness* (i.e. the extent to which the individuals they represent are suitable to the environment). Using their fitness as a criterion, certain chromosomes in the population are selected for reproduction based on fitness (survival of the fittest). These selected individuals are manipulated by crossover and mutation operators. The crossover operator is applied to a pair of selected parents to create offspring, and the mutation operator is used as a slight modification to this offspring, or of the remaining members of the population. The repeated application of the genetic operators to the fittest chromosomes results in an increase in the average fitness of the population over time and thus to the identification of improved solutions to the problem under investigation. Each new generation of solutions is seen to be in some sense an improvement over the previous one. This process is iteratively repeated until there appears in the population an acceptable solution to the problem at hand. This is summarized as given below.

Structure of a Simple GA

```
SimpleGeneticAlgorithm{
    initialize population;
    evaluate population;
    While satisfactory solution not located
    {
        scale population fitness;
        select solutions for next generation;
        perform crossover and mutation;
        evaluate population;
    }
}
```

Several methods had been proposed to handle constraints with GAs [40,28,17,3]. Many adaptive penalty function schemes for GAs have been discussed in [3,31,8]. An Extended Hybrid Genetic Algorithm (EHGA), that makes use of the gradient direction of the objective function was proposed by Fung [18]. Mitsuo Gen et.al. [24,51] and Lino Costa et.al. [7] proposed a method for solving nonlinear mixed integer programming problems using genetic algorithms and multiplicative penalty function. They used the mean of satisfactory degrees of system constraints for constructing the penalty function. Here we make use of a modified form of the evaluation function for stochastic optimizations. Some disadvantages with GA are:

- Difficult to understand and explain to end users.
- Abstraction of the problem and method to represent individuals is quite difficult.
- Determining fitness function is difficult.
- Determining how to perform crossover and mutation is difficult.
- Tuning of GA parameters takes significant time.
- GAs are computationally very intensive and the optimum solution is not guaranteed.

4.2 Approaches for Generating Candidate Solutions

There are several approaches proposed in genetic algorithms to handle constraint optimisation problems. They can be divided into four categories:

- Discarding infeasible solutions,
- Preserving feasibility of solutions, either from a special representation method or from some special genetic operators,
- Repairing infeasible solutions to feasible ones,
- Penalizing infeasible solutions and making solutions adhere more and more to feasible regions of search space.

Each method has its own advantages and disadvantages. And there are some hybrid methods that apply more than one of those approaches to balance the strong/weak points of them.

Discard Infeasible Solutions In this approach we simply discard infeasible solutions and continue recombination and mutation until a feasible solution is produced. This approach has the advantage that there would never be any infeasible solutions in the population. But it also has the disadvantage that feasible solutions may be difficult to find, because it can spend a great deal of time in the evaluation and rejection of those infeasible solutions, especially when the problem is highly constrained. Further more, it considers no points outside the feasible regions of the search space, which could be bad. Because constraint management techniques allowing movements through infeasible regions of the search space tend to yield more rapid optimization and produce better final solutions than do approaches limiting search trajectories only to feasible regions of the search space. And for cases when optimal solutions lie on the boundaries of feasible regions, this approach will not work well.

Preserve feasibility of solutions Preserving feasibility of solutions usually requires problem specific chromosome representation and/or genetic operators. One approach of preserving feasibility is to combine the probabilistic mechanisms essential in classical GAs with heuristics common in traditional constraint satisfaction problems. The resulting algorithm is often called Heuristic Genetic Algorithm (HGA). In HGA, the classical uniform random mechanism of the crossover operator is replaced by a deterministic or probabilistic mechanism based on heuristics, which takes into account some of the prescribed constraints. This special heuristic genetic operator is created to counterbalance feasibility and randomness. Namely, the application of heuristics improves the performance of the blind random mechanism and the random component can compensate the strong bias introduced by the heuristics. One big problem with HGA is that although adding heuristics to GAs will improve performance in the first few generations, the performance will degrade rapidly due to the fact that heuristic crossovers tend to decrease genetic diversity much faster than non-heuristic ones. Same as the first category, this kind of solutions has the advantage of never or rarely generate infeasible solutions, but it suffered from difficulties in finding the feasible solutions or inefficiency in reaching the optimal solution. And being problem specific is another major disadvantage of this approach.

Repair Infeasible Solutions This strategy depends on the existence of a rapid, deterministic repair procedure for converting a solution that violates

hard constraints into one that does not. There are two kinds of repair process for infeasible solutions:

- (1) To replace infeasible chromosomes with their repaired counterparts generated for purposes of evaluation,
- (2) To use the repair strategies to determine the evaluation of a solution, and not used to modify the genetic composition of the chromosome itself.

In practice, replacing original solutions with their repaired counterpart works well when solutions in the feasible regions are better at the center than at the edges. And non-replacement method works well when there are many disjoint feasible regions because it allows the optimization process to wander widely in the search space so that movement from basin to basin can occur. However, if the best solutions tend to lie at the center of the feasible regions, then the algorithm might never wander close enough to the feasible region to find them. There are a great many factors that will impact the performance of algorithms employing these repair strategies. Such factors include quality of the repair mechanism; the amount of CPU time required to generate a repaired solution; the gradient of the search space outside the feasible regions; and the diversity and nature of the mapping from infeasible points to feasible ones.

Penalize Infeasibility Penalty functions have been used to transform a constrained optimization problem into an unconstrained optimization problem. There are two kinds of commonly used penalty functions for infeasible solutions. One is to launch a uniform penalty function for infeasibility during all iterations. The other is to launch negligible penalties for infeasibility at the beginning of the optimization process and with such penalties gradually becoming larger and larger. The later approach seemed to be a better one because it allows the GA to explore more of the searching space at the beginning. And by increasing the penalty proportionally to the generation number, it increases the pressure for the population to generate more feasible solutions later on and converge at the end. Also, it has been shown that penalty functions based on the distance from feasibility outperform those based on the number of violated constraints.

Penalizing infeasible solutions has the advantage that it is able to consider infeasible solutions. But, it has the disadvantage that it may never generate feasible solutions: the initial negligible penalties for infeasibility may lead the algorithm to regions far from feasible solutions – regions in which the search process may be stranded in infeasible local optima as penalties for infeasibility increase. And applying penalty to the infeasible solutions requires a careful selection of control parameters (like a degree to which each constraint is penalized). If one imposes a high degree of penalty, more emphasis is placed on obtaining feasibility and the GA will move very quickly towards a feasible solution. The system will tend to converge to a feasible point even if it is far from optimal. However, if one imposes a low degree of penalty, less emphasis is placed on feasibility, and the system may never converge to a feasible solution.

Furthermore, penalty functions are often computationally expensive in more constrained problems.

4.3 Computational Framework Combining GA and Monte Carlo Simulation

This section focuses on solving a stochastic optimization problem using genetic algorithm and Monte Carlo simulation. The outline of the optimization system is given in figure 4. The stochastic and deterministic functions that defines the optimization problem is passed to the GA module. If the function is stochastic, the GA module will invoke the Monte Carlo module to get an approximate estimate of the function value and for deterministic case simulation is not done.

To design the GA, we choose a convenient problem representation. Let the inequality constraints given in the formulation be mapped into $G_i(x) \leq 0, i = 1, 2, \dots, p$ and the equality constraints be mapped into $H_i(x) = 0, i = 1, 2, \dots, q$ to obtain a deterministic equivalent of the formulation given in 2. Now the model to be solved using GA is

$$\min: E[f(\mathbf{x}, \xi)] \quad (59)$$

or

$$\max: P[f(\mathbf{x}, \xi) \leq u] \quad (60)$$

subject to

$$G_j(\mathbf{x}) \leq 0, j = 1, 2, \dots, p \quad (61)$$

$$H_j(\mathbf{x}) = 0, j = 1, 2, \dots, q \quad (62)$$

$$x_j^L \leq x_j \leq x_j^U, j = 1, 2, \dots, n \quad (63)$$

The GA is designed for the above problem as given below:

Penalty Function The GAs usually optimizes an unconstrained problem. In this algorithm a penalty function is used to transform the constrained optimization problem into an unconstrained one. There are several methods for constraint handling within the genetic algorithm framework. Additive penalty method (APM)[17,3,51,7], is a commonly used penalty function approach in which a penalty cost proportional to the total constraints violation is added to the objective function. Multiplicative penalty method (MPM)[24], multiplies the objective function by a factor proportional to the total constraints violation. Here an adaptive MPM is used for the evaluation function of the GA by introducing the degree of constraints satisfactory in order to solve constrained optimization problems. The various notations are defined as follows.

Define the violation of system constraints as follows:

$$\begin{aligned} [G_i(\mathbf{x})]_+ &= \max\{0, G_i(\mathbf{x})\}, \quad i = 1, 2, \dots, p \\ [H_i(\mathbf{x})]_+ &= \max\{0, H_i(\mathbf{x})\}, \quad i = 1, 2, \dots, q \\ [H_i(\mathbf{x})]_- &= \min\{0, H_i(\mathbf{x})\}, \quad i = 1, 2, \dots, q \end{aligned} \quad (64)$$

Define the maximum violation of constraints in current population as follows:

$$\begin{aligned} [G_i(\mathbf{x})]_+^{max} &= \max_k \{[G_i(\mathbf{x})]_+\}, \quad i = 1, 2, \dots, p \\ [H_i(\mathbf{x})]_+^{max} &= \max_k \{[H_i(\mathbf{x})]_+\}, \quad i = 1, 2, \dots, q \\ [H_i(\mathbf{x})]_-^{min} &= \min_k \{[H_i(\mathbf{x})]_-\}, \quad i = 1, 2, \dots, q \end{aligned} \quad (65)$$

where $k = 1, \dots, popsize$ where $popsize$ is the current population size.

Now introduce the degree of constraint satisfactory, $\alpha_i(G_i(\mathbf{x}))$ for inequality constraints and $\alpha_i(H_i(\mathbf{x}))$ for equality constraints are evaluated in the following way: 1. For each inequality constraints,

$$\alpha_i(G_i(\mathbf{x})) = \begin{cases} 1 & \text{if } G_i(\mathbf{x}) \leq 0; \\ \frac{[G_i(\mathbf{x})]_+^{max} - [G_i(\mathbf{x})]_+}{[G_i(\mathbf{x})]_+^{max}} & \text{if } G_i(\mathbf{x}) > 0. \end{cases} \quad i = 1, 2, \dots, p \quad (66)$$

2. For each equality constraints,

$$\alpha_i(H_i(\mathbf{x})) = \begin{cases} 1 & \text{if } H_i(\mathbf{x}) = 0; \\ \frac{[H_i(\mathbf{x})]_+^{max} - [H_i(\mathbf{x})]_+}{[H_i(\mathbf{x})]_+^{max}} & \text{if } H_i(\mathbf{x}) > 0; \\ \frac{[H_i(\mathbf{x})]_-^{min} - [H_i(\mathbf{x})]_+}{[H_i(\mathbf{x})]_-^{min}} & \text{if } H_i(\mathbf{x}) < 0. \end{cases} \quad i = 1, 2, \dots, q \quad (67)$$

The degree of satisfaction of constraints α is illustrated in figures 2 and 3. Using the degrees of constrained satisfaction, we create the penalty function $P(\gamma, \mathbf{x})$ for the maximizing problem, for which $P(\gamma, \mathbf{x}) = 1$, if the candidate solution is feasible, and if the candidate solution violates all constraints in maximum way, $P(\gamma, \mathbf{x}) = 0$, as follows:

$$P(\gamma, \mathbf{x}) = \left\{ \frac{1}{p+q} \left[\sum_{i=1}^p \alpha_i(G_i(\mathbf{x})) + \sum_{i=1}^q \alpha_i(H_i(\mathbf{x})) \right] \right\}^\gamma \quad (68)$$

Thus for maximizing case $0 \leq P(\gamma, \mathbf{x}) \leq 1$.

For minimizing problem, the penalty function $P(\gamma, \mathbf{x})$ is created to be $P(\gamma, \mathbf{x}) = 1$, if the candidate solution is feasible, otherwise $1 < P(\gamma, \mathbf{x}) \leq 2^\gamma$, as follows:

$$P(\gamma, \mathbf{x}) = \left\{ 2 - \frac{1}{p+q} \left[\sum_{i=1}^p \alpha_i(G_i(\mathbf{x})) + \sum_{i=1}^q \alpha_i(H_i(\mathbf{x})) \right] \right\}^\gamma \quad (69)$$

Multiplication of satisfactory degrees of all system constraints can also be used for constructing the penalty function. In the case of maximization penalty $P(\gamma, \mathbf{x})$ is:

$$P(\gamma, \mathbf{x}) = \left\{ \left[\prod_{i=1}^p \alpha_i(G_i(\mathbf{x})) \cdot \prod_{i=1}^q \alpha_i(H_i(\mathbf{x})) \right] \right\}^{\gamma} \quad (70)$$

In the case of minimization penalty $P(\gamma, \mathbf{x})$ becomes:

$$P(\gamma, \mathbf{x}) = \left\{ 2 - \left[\prod_{i=1}^p \alpha_i(G_i(\mathbf{x})) \cdot \prod_{i=1}^q \alpha_i(H_i(\mathbf{x})) \right] \right\}^{\gamma} \quad (71)$$

where $\gamma \geq 1$ is called penalty parameter.

Evaluation Function The penalty is multiplied with the objective function value to obtain the fitness value of the solution. The mean and the product of satisfactory degrees of all system constraints are used for construction of the evaluation function. Both of them can be used for creating the penalty term. The penalty is multiplied with the normalized objective function value (or to a power of this value) to obtain the fitness value of the solution. Thus the

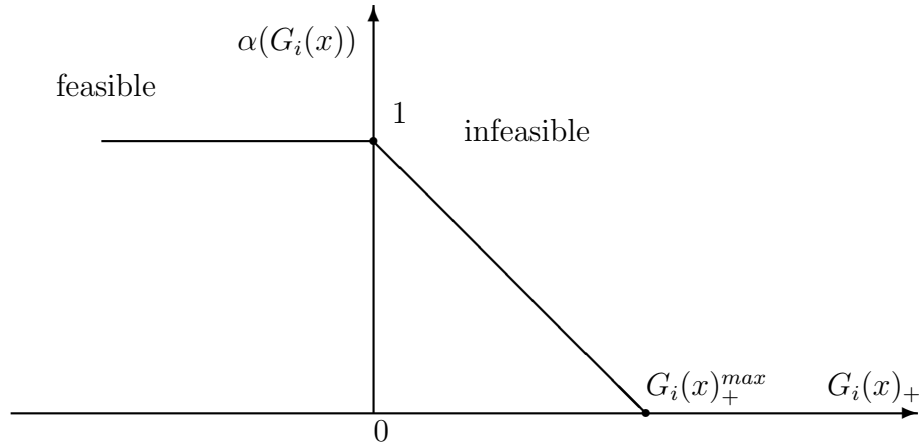


Fig. 2. Degree of satisfaction of inequality constraints in GA

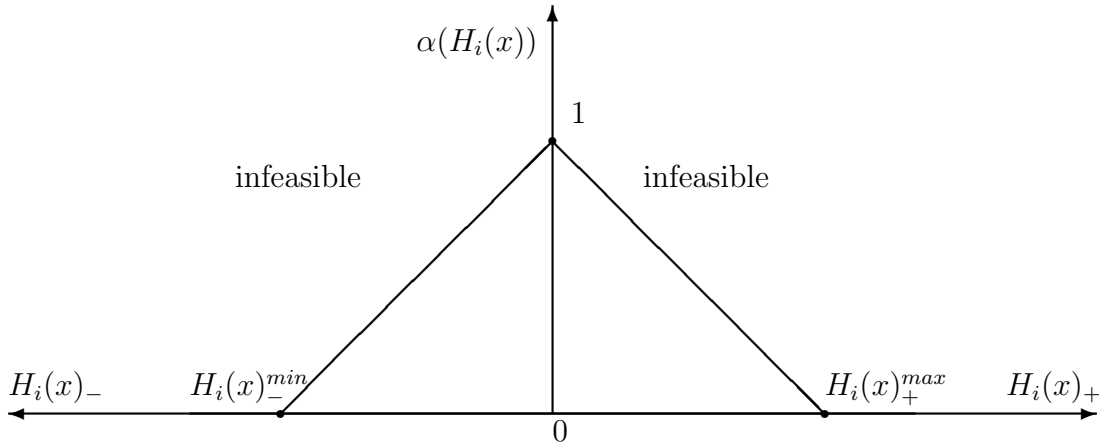


Fig. 3. Degree of satisfaction of equality constraints in GA

evaluation function (fitness function) of GA for a solution \mathbf{x} is constructed as follows:

$$F(\mathbf{x}) = e^{a \frac{E[f(\mathbf{x}, \xi)] - E[f(\mathbf{x}, \xi)]^{min}}{E[f(\mathbf{x}, \xi)]^{max} - E[f(\mathbf{x}, \xi)]^{min}}} . P(\gamma, \mathbf{x}) \quad (72)$$

For probability maximization it is

$$F(\mathbf{x}) = e^{a \frac{P[f(\mathbf{x}, \xi) \leq u] - P[f(\mathbf{x}, \xi) \leq u]^{min}}{P[f(\mathbf{x}, \xi) \leq u]^{max} - P[f(\mathbf{x}, \xi) \leq u]^{min}}} . P(\gamma, \mathbf{x}) \quad (73)$$

Where a is a nonnegative control parameter and superscripts *max* and *min* indicates the maximum and minimum respectively in the current population. The normalized values of the objective function of the individuals in the population ranges from 0 to 1. Thus the best fitness value an individual can have is 1 in the case of minimization. The fitness values of the individuals in a population can vary from 0 to ∞ .

Chromosome Representation A Real valued representation scheme is used for solutions. Thus, an individual \mathbf{V}_k , the k -th chromosome of the population, is represented as follows:

$$\mathbf{V}_k = [x_1^k, \dots, x_n^k] \text{ where } k = 1, \dots, \text{popsize},$$

with $x_i^k \in \mathbb{R}$ ($i = 1, \dots, n$).

Initial Population Individuals $\mathbf{V}_k = [x_1^k, \dots, x_n^k]$ ($k = 1, \dots, \text{popsize}$) of the first population are generated by uniform random sampling, with $l_i \leq x_i^k \leq u_i$, where $[l_i, u_i]$ ($i = 1, \dots, n$)

Selection of Parents In proportional selection, the individuals are selected according to their relative fitness values. The selection probability of k -th individual at generation g is given by

$$p_s(\mathbf{V}_k^g) = \frac{F(\mathbf{V}_k^g)}{\sum_{k=1}^{\text{popsize}} F(\mathbf{V}_k^g)} \quad (74)$$

In proportional selection, the individuals are selected according to their relative fitness values. Proportional selection is usually implemented with a Monte

Carlo or roulette wheel selection. The pseudo code is given below.

Proportional selection

Input: The Population $P(g)$

Output: A pair of chromosomes $\mathbf{V}'_1, \mathbf{V}'_2$

Exponential $(\mathbf{V}_1, \dots, \mathbf{V}_{popsize})$

$s_0 \leftarrow 0$

for $i \leftarrow 1$ to $popsize$ do

$s_i \leftarrow s_{i-1} + p_s(\mathbf{V}_i)$

endfor

for $i \leftarrow 1$ to 2 do

$r \leftarrow random[0, s_{popsize}]$

$\mathbf{V}'_i \leftarrow \mathbf{V}_l$ such that $s_{l-1} \leq r < s_l$

endfor

return $\{\mathbf{V}'_1, \mathbf{V}'_2\}$

In rank based selection, the population is sorted from best to worst. Rank 1 is assigned to the best and rank $popsize$ is assigned to the worst individual. The probability that an individual will be selected for reproduction is inversely proportional to the rank assigned to it. In exponential ranking selection, the probabilities of the ranked individuals are exponentially weighted. Thus, the selection probability of k -th individual at generation g is given by

$$p_s(\mathbf{V}_k^g) = \frac{c^{(rank_k-1)}}{\sum_{k=1}^{popsize} c^{(rank_k-1)}} = \frac{1-c}{1-c^{popsize}} c^{(rank_k-1)} \quad (75)$$

where c , $0 < c < 1$ is a probability distribution parameter that controls the distribution used for choosing parents. The procedure is given below.

Exponential ranking selection

```

Input: The Population  $P(g)$ 
Output: A pair of chromosomes  $\mathbf{V}'_1, \mathbf{V}'_2$ 
Exponential  $(\mathbf{V}_1, \dots, \mathbf{V}_{popsize})$ 
 $s_0 \leftarrow 0$ 
for  $i \leftarrow 1$  to  $popsize$  do
     $s_i \leftarrow s_{i-1} + p_s(\mathbf{V}_i)$ 
endfor
for  $i \leftarrow 1$  to  $2$  do
     $r \leftarrow random[0, 1]$ 
     $\mathbf{V}'_i \leftarrow \mathbf{V}_l$  such that  $s_{l-1} \leq r < s_l$ 
endfor
return  $\{\mathbf{V}'_1, \mathbf{V}'_2\}$ 

```

Genetic Operators *Arithmetic crossover* operator (which is defined as a linear combination of two chromosome) is used. With arithmetic crossover, given two chromosomes selected for crossover in generation g , \mathbf{V}_j^g and \mathbf{V}_l^g , we obtain two offsprings by means of linear recombination of their parents as follows:

$$\mathbf{V}_j^{g+1} = c\mathbf{V}_j^g + (1 - c)\mathbf{V}_l^g \quad (76)$$

$$\mathbf{V}_l^{g+1} = c\mathbf{V}_l^g + (1 - c)\mathbf{V}_j^g \quad (77)$$

$j, l \in 1, 2, \dots, popsize$ where c is a random number in the range $[0, 1]$.

Uniform mutation changes a single component of an individual. Given an individual $\mathbf{V}_k = [x_1^k, \dots, x_i^k, \dots, x_n^k]$, uniform mutation generates an offspring $\mathbf{V}'_k = [x_1^k, \dots, x_i'^k, \dots, x_n^k]$ where $x_i'^k$ is a random value from the domain of the random variable.

Nonuniform mutation is used for manipulating non-integer variables. The action of non-uniform mutation depends on the age of the population, and its effect is a local fine tuning in the last generations of the GA. Given an individual $\mathbf{V}_k = [x_1^k, \dots, x_i^k, \dots, x_n^k]$, nonuniform mutation generates an offspring $\mathbf{V}'_k = [x_1^k, \dots, x_i'^k, \dots, x_n^k]$ with

$$x_i'^k = \begin{cases} (1 - p)x_i^k + px_i^L & \text{with probability } 1/2 \\ (1 - p)x_i^k + px_i^H & \text{with probability } 1/2 \end{cases}$$

Where $[x_i^L, x_i^H]$ is the domain of the variable, $i \in \{1, 2, \dots, n\}$ chosen randomly and $p = (1 - g/G)^B u$, g is the current generation number, G is the maximum number of generations, $B > 0$ is a tuning parameter and $u \sim U(0, 1)$.

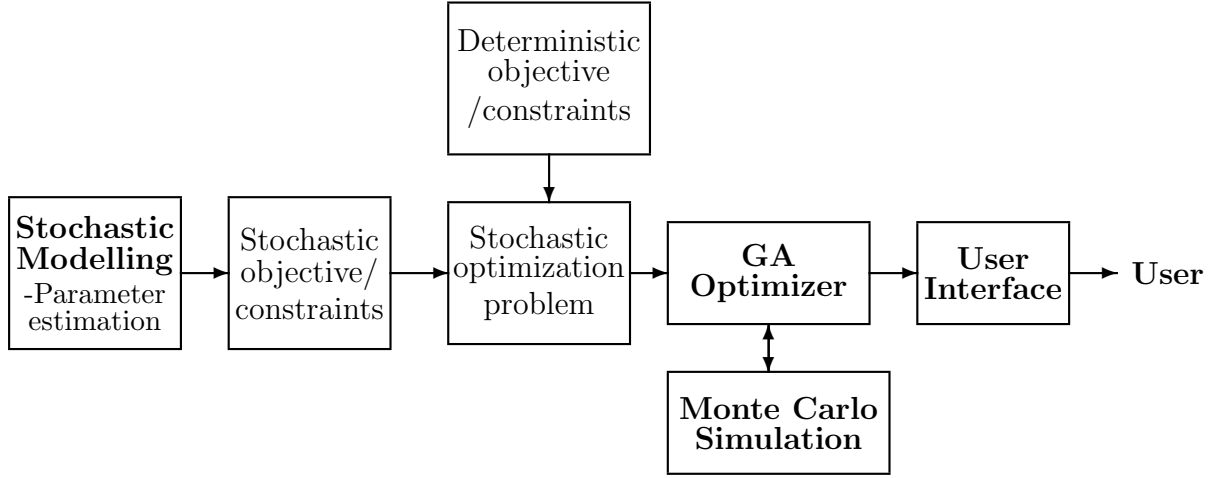


Fig. 4. Stochastic optimization using GA and Simulation

4.4 Procedure

Step 1: Set population size $popsiz$ e, crossover probabilities p_c , mutation probability p_{m1}, p_{m2} , maximum generation G , current population $g = 0$, the penalty parameter γ , the control parameter a , and the constant value B for the non-uniform mutation.

Step 2: Generate initial population $\mathbf{V}_k^0 (k = 1, \dots, popsiz$ e)

Step 3: Use Monte Carlo Simulation to obtain the expected values and probabilities in constraints for each chromosome in the population. Compute the objective function values by Monte Carlo Simulation. Determine the fitness value $F(\mathbf{V}_k)$, of each chromosome as described earlier.

Step 4: Select parents from $P(g)$ by using a selection strategy. Select elites from $P(g)$ for the next generation. Perform crossover and mutation to generate a new population $P(g + 1)$. Set $g = g + 1$.

Step 5: If $g < G$, goto **Step 3**. If $g = G$, output the chromosome with the best feasible solution and terminate.

4.5 Evaluation of Expected Values and Probabilities

Available methods for evaluating the expected values and probabilities are Monte Carlo Simulation (MCS), first order reliability method (FORM), second

order reliability method (SORM), and the family of advanced mean-value methods. In order to get the mean value and standard deviation of $f(\mathbf{x}, \xi)$, the objective function can be expanded using Taylor series around the mean value of the random variables (ξ_i), and only linear terms in the Taylor series are considered as below:

$$f(\mathbf{x}, \xi) \approx f(\mathbf{x}, \bar{\xi}) + \sum_{i=1}^m (\xi_i - \bar{\xi}_i) \left. \frac{\partial f}{\partial \xi_i} \right|_{\xi=\bar{\xi}} \quad (78)$$

where $\bar{\xi}_i$ and $\bar{\xi}$ denote the expected values of the random variable $\bar{\xi}$ and random vector ξ respectively. The mean and standard deviation of $f(\mathbf{x}, \xi)$ can be given as:

$$\begin{aligned} E[f(\mathbf{x}, \xi)] &\approx f(\mathbf{x}, \bar{\xi}) \\ Var[f(\mathbf{x}, \xi)] &\approx \sum_{i=1}^m \sum_{j=1}^m \left. \frac{\partial^2 f}{\partial \xi_i \partial \xi_j} \right|_{\xi=\bar{\xi}} Cov(\xi_i, \xi_j) \end{aligned} \quad (79)$$

Monte Carlo Simulation Suppose we need to evaluate a multi-dimensional definite integral of the form

$$I_n = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \dots \int_{a_n}^{b_n} f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n \quad (80)$$

Generate random numbers x_i with uniform probability density function given by

$$f_i(x_i) = \begin{cases} \frac{1}{b_i - a_i} & \text{if } a_i \leq x_i \leq b_i; \\ 0 & \text{otherwise.} \end{cases} \quad (81)$$

then assuming x_i are independent

$$I_n = \prod_{i=1}^n (b_i - a_i) E[f(x_1, x_2, \dots, x_n)] \quad (82)$$

$$\approx \frac{D}{N} \sum_{j=1}^N f(x_1^j, x_2^j, \dots, x_n^j) \quad (83)$$

is a random variable whose expectation is the value of the integral and whose standard deviation is $O(N^{-1/2})$, D is the volume of the bounded region. In contrast to the classical method of solving multiple integrals by iteration, the number of sampling points to obtain a given degree of accuracy is independent of the number of dimensions, which makes Monte Carlo simulation very attractive for very large number of dimensions. The time complexity of the problem will be the sum of time spent for Monte Carlo Simulation and the time spent for the evolution program. The time spent for Monte Carlo simulation is exactly proportional to the number of sampling points. Time complexity of

the evolution program increases linearly with the number of decision variables. Thus the expected values of the form $E[f(\mathbf{x}, \xi)] = \int_{\mathbb{R}^m} f(\mathbf{x}, \xi)\phi(\xi)d\xi$ can be estimated as $\frac{D}{N} \sum_{i=1}^N f(\mathbf{x}, \xi_i)\phi(\xi_i)$.

The application of the Monte-Carlo procedure usually requires 1000-2000 trials for statistical simulation and estimation of one value of the function. The iterative method used here to solve the deterministic optimization problem resulting from sampling is based on an evolutionary algorithm. In this, a new sample is generated at every iteration rather than fixing a sample from the beginning. A study of *variable-sample* techniques for stochastic optimization problems is given in Tito [14]. The procedure is as given below.

Procedure MCS for Expected Value

```

for  i = 1  to  number_simulation  do
     $\xi^i = \text{vector\_random}()$ ;
     $\text{objective+} = f(\mathbf{x}, \xi^i)\phi(\xi^i)$ ;
endfor
 $\text{objective*} = D/\text{number\_simulation}$ ;

```

For any given \mathbf{x} the following Monte Carlo technique can be used to estimate probabilities of the form $P[g_j(\mathbf{x}, \xi) \leq 0]$ and $P[h_j(\mathbf{x}, \xi) = 0]$. We generate independent random vectors ξ , from their probability distributions, where the generating methods have been well-discussed by numerous literatures and summarized by [42]. Let N' be the number of occasions on which $g_j(\mathbf{x}, \xi) \leq 0$ or $h_j(\mathbf{x}, \xi) = 0$, ie. Number of random vectors satisfying the constraints. Then by the basic definition of probability, $P[g_j(\mathbf{x}, \xi) \leq 0]$ or $P[h_j(\mathbf{x}, \xi) = 0]$ can be estimated by

$$P[g_j(\mathbf{x}, \xi) \leq 0] \quad \text{or} \quad P[h_j(\mathbf{x}, \xi) = 0] = N'/N \quad (84)$$

Certainly , this estimation is approximate and may change from a simulation to another. But it is available to real practice problem since the determination of the confidence level α or β itself is not precise. The procedures are as given below.

Procedure MCS for Probability

```

for  $i = 1$  to  $number\_simulation$  do
     $\xi^i = vector\_random()$ ;
    if  $g(\mathbf{x}, \xi^i) \leq 0$ 
         $probability+ = 1$ ;
    endifor
 $probability/ = number\_simulation$ ;
for  $i = 1$  to  $number\_simulation$  do
     $\xi^i = vector\_random()$ ;
    if  $h(\mathbf{x}, \xi^i) = 0$ 
         $probability+ = 1$ ;
    endifor
 $probability/ = number\_simulation$ ;

```

The probability distribution and probability density function of the stochastic vector \mathbf{x} are $\Phi(\xi)$ and $\phi(\xi)$ respectively.

Importance Sampling

Various reduction techniques have been proposed in order to improve the accuracy and efficiency of the MCS method. Importance sampling (IS) is recognized as the most efficient reduction technique [54]. The key-idea of this technique is to select a non-negative importance sampling density function $f_s(\xi)$. Instead of choosing points from a uniform distribution, they are now chosen from a distribution which concentrates the points where the function being integrated is large.

$$P [g_j(\mathbf{x}, \xi) \leq 0] = \int_{g_j(\mathbf{x}, \xi) \leq 0} f_\xi(\xi) d\xi = \int_{g_j(\mathbf{x}, \xi) \leq 0} \frac{f_\xi(\xi)}{f_s(\xi)} f_s(\xi) d\xi \quad (85)$$

An unbiased estimator of probability is given by

$$P [g_j(\mathbf{x}, \xi) \leq 0] = \frac{1}{N} \sum_{i=1}^N I(\xi^i) \frac{f_\xi(\xi^i)}{f_s(\xi^i)} \quad (86)$$

Each ξ^i is generated using $f_s(\xi^i)$. $I(\xi^i)$ is an indicator for the successful and unsuccessful simulations defined as

$$I(\xi^i) = \begin{cases} 1 & \text{if } g_j(\mathbf{x}, \xi^i) \leq 0 \\ 0 & \text{if } g_j(\mathbf{x}, \xi^i) > 0 \end{cases} \quad (87)$$

In stochastic optimization, the stochastic module is required to run with a large number of iterations at each optimization stage. So the computational

intensity of this stochastic optimization method is large. In order to circumvent this, a novel delayed sampling approach is proposed in [19].

First-Order Reliability Method (FORM) FORM analysis presumes that the mean and covariance are sufficient to characterize the distribution of the dependent variables, i.e., the model results are assumed to be normally distributed. Therefore FORM is a computationally more efficient approach. The original random parameter is transformed into an independent and standard normal random parameter. We have

$$P [g_j(\mathbf{x}, \xi) \leq 0] = \int_{g_j(\mathbf{x}, \xi) \leq 0} \dots \int f_\xi(\xi_1, \xi_2, \dots, \xi_m) d\xi_1 d\xi_2 \dots d\xi_m \quad (88)$$

$f_\xi(\xi)$ is the joint probability density function of all random parameters. Constraints $g_j(\mathbf{x}, \xi) \leq 0$ are generally nonlinear. To overcome this difficulty, the constraint function is expanded as a Taylor series at the mean values of the random variables and only the linear part is taken into account. Thus the above integral can be approximated as:

$$P [g_j(\mathbf{x}, \xi) \leq 0] \approx \Phi \left\{ -\frac{\bar{g}_j(\mathbf{x}, \xi)}{\sigma[g_j(\mathbf{x}, \xi)]} \right\} \quad (89)$$

where

$$\begin{aligned} \bar{g}_j(\mathbf{x}, \xi) &\approx g_j(\mathbf{x}, \bar{\xi}) \\ Var [g_j(\mathbf{x}, \xi)] &\approx \sum_{i=1}^m \sum_{k=1}^m \left. \frac{\partial^2 g_j}{\partial \xi_i \partial \xi_k} \right|_{\xi=\bar{\xi}} Cov(\xi_i, \xi_k) \end{aligned} \quad (90)$$

If the constraints are highly nonlinear and the random variables follow non-identical distributions, the results obtained by FORM are inaccurate. Hence we use Monte carlo simulation here.

4.6 Computational Results

In this research effort, a JAVA GA framework was developed to solve non-convex nonlinear optimization problems with deterministic and stochastic parameters. This framework includes classes for GA (generational GA with elitism and rank-based selection), GA operators (these result in what we call standard GA), Monte-Carlo simulation, packages for generating random numbers and classes for defining the problem and constraints. JAVA's features such as portability, robustness, object-orientation outweigh its degraded performance as compared to other compiled languages like C, C++. The advantages of JAVA are particularly evident in the area of distributed computing. The disadvantage with this algorithm is that it takes significant time for fine-tuning the parameters for genetic algorithms.

The performance characteristics of this algorithm using different operators and

control parameters have been investigated extensively on several optimization problems from literature. A number of independent runs were executed for each test problem. We have tested the proposed methodology on NLP and CCP models. **Problem I** Fractional programming

$$\max \quad f(\mathbf{x}) = \frac{3x_1 + x_2 - 2x_3 + 0.8}{2x_1 - x_2 + x_3} + \frac{4x_1 - 2x_2 + x_3}{7x_1 + 3x_2 - x_3}$$

subject to

$$\begin{aligned} x_1 + x_2 - x_3 &\leq 1, \\ -x_1 + x_2 - x_3 &\leq -1, \\ 12x_1 + 5x_2 + 12x_3 &\leq 34.8, \\ 12x_1 + 12x_2 + 7x_3 &\leq 29.1, \\ -6x_1 + x_2 + x_3 &\leq -4.1, \\ x_i &\geq 0, i = 1, 2, 3. \end{aligned}$$

The known global optimum is $\mathbf{X}^* = (1, 0, 0)$ with $f(\mathbf{X}^*) = 2.471428$. A run of 500 generations gave the optimum as $f(\mathbf{X}^*) = 2.47142853$ at $\mathbf{X}^* = (0.999999966, 8.778\text{E-}11, 3.374\text{E-}8)$ SNOPT 6.1 gives $f(\mathbf{X}^*) = 2.471428571$ at $\mathbf{X}^* = (1, 0, 0)$.

Problem II

$$\min \quad f(\mathbf{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$$

subject to nonlinear constraints

$$\begin{aligned} (x_1 - 5)^2 + (x_2 - 5)^2 - 100 &\geq 0.0, \\ -(x_1 - 6)^2 - (x_2 - 5)^2 + 82.81 &\geq 0.0, \\ 13 &\leq x_1 \leq 100, \\ 0 &\leq x_2 \leq 100, \end{aligned}$$

The known global optimum is $\mathbf{X}^* = (14.095, 0.84296)$ with $f(\mathbf{X}^*) = -6961.81381$. A run of 2000 generations gave the optimum as $f(\mathbf{X}^*) = -6960.2895$ at $\mathbf{X}^* = (14.0957, 0.8443)$ SNOPT 6.1 gives $f(\mathbf{X}^*) = -6961.813876$ at $\mathbf{X}^* = (14.095, 0.842961)$.

Problem III The third example was to maximize a multimodal function on a convex set.

$$\max \quad f(\mathbf{x}) = \sum_{i=1}^n x_i \cdot \sin(i\pi x_i)$$

subject to

$$\sum_{i=1}^n x_i^2 \leq 100,$$

$$0 \leq x_i \leq 20, i = 1, 2, 3, \dots, 7$$

For $n=7$, a run of 1000 generations gave the optimum as $f(\mathbf{X}^*) = 26.01025$ at $\mathbf{X}^* = (4.5033, 4.2248, 3.5026, 3.6251, 4.101, 3.7507, 2.3579)$. Slack in constraint is 0.016. This is a better optimum than the results reported in M.Gen et al., 1996 using hybrid intelligent algorithm. SNOPT 6.1 finds a local optimum for $x_i \geq 2, i = 1, \dots, 7$. $f(\mathbf{X}^*) = 17.13234718$ at $\mathbf{X}^* = (2.53969, 2.26118, 2.17185, 2.62741, 2.50162, 2.41783, 2.64364)$.

Problem IV This example is a stochastic optimization with three decision variables and three stochastic variables. Consider the stochastic expected value model

$$\min \quad E \left[\sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2 + (x_3 - \xi_3)^2} \right]$$

subject to deterministic constraint

$$x_1^2 + x_2^2 + x_3^2 \leq 10$$

Where ξ_1 is $\mathcal{U}(1, 2)$, ξ_2 is $\mathcal{N}(3, 1)$ and ξ_3 is $\mathcal{E}\mathcal{X}\mathcal{P}(4)$. A run of the genetic algorithm (200 generations in GA, 1000 simulations) shows that the optimal solution is $\mathbf{X}^* = (1.4, 2.8, 0.15)$ with $f(\mathbf{X}^*) = 0.91$

Chance Constrained Models Problem V This is an abstract numerical example in which the objective function is multimodal and highly nonlinear.

$$\max \quad f(\mathbf{x}) = \sum_{i=1}^3 x_i \cdot \sin(i\pi x_i)$$

subject to

$$P[\xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 \leq 10] \geq 0.70,$$

$$P[\eta_1 x_1^2 + \eta_2 x_2^2 + \eta_3 x_3^2 \leq 100] \geq 0.80,$$

$$x_1, x_2, x_3 \geq 0.$$

Where ξ_1 is $\mathcal{U}(0.8, 1.2)$, ξ_2 is $\mathcal{U}(1, 1.3)$, ξ_3 is $\mathcal{U}(0.8, 1.0)$, η_1 is $\mathcal{N}(1, 0.5)$, η_2 is $\mathcal{E}\mathcal{X}\mathcal{P}(1.2)$ and η_3 is $\mathcal{L}\mathcal{O}\mathcal{N}\mathcal{G}(0.8, 0.6)$. A run of the genetic algorithm (100 generations in GA, 500 simulations) shows that the global optimal solution is $\mathbf{X}^* = (2.02, 1.01, 6.82)$ with $f(\mathbf{X}^*) = 9.5444$. This solution, achieved with a lesser number of generations, is better than the results (3000 generations, objective = 9.3537) reported in [25]. **Problem VI** Consider the following chance constrained programming problem with four decision variables and nine stochastic parameters.

$$\max \quad x_4$$

subject to chance constraints

$$\begin{aligned}
P [\xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 \geq x_4] &\geq 0.90, \\
P [\eta_1 x_1^2 + \eta_2 x_2^2 + \eta_3 x_3^2 \leq 8] &\geq 0.80, \\
P [\tau_1 x_1^3 + \tau_2 x_2^3 + \tau_3 x_3^3 \leq 15] &\geq 0.85, \\
x_1, x_2, x_3 &\geq 0.
\end{aligned}$$

Where ξ_1, η_1 , and τ_1 are $\mathcal{U}(1, 2)$, $\mathcal{U}(2, 3)$, and $\mathcal{U}(3, 4)$ respectively. ξ_2, η_2 , and τ_2 are $\mathcal{N}(1, 1)$, $\mathcal{N}(2, 1)$, and $\mathcal{N}(3, 1)$ respectively. And ξ_3, η_3 , and τ_3 are $\mathcal{E}\mathcal{X}\mathcal{P}(1)$, $\mathcal{E}\mathcal{X}\mathcal{P}(2)$, and $\mathcal{E}\mathcal{X}\mathcal{P}(3)$ respectively. A run of the hybrid algorithm (500 generations in GA, 500 simulations) shows that the optimal solution is $\mathbf{X}^* = (1.52, 0.39, 1.48, 2.53)$ with $f(\mathbf{X}^*) = 2.53$ **Problem VII Feed Mixer Problem** Van de Panne and Popp [34] presented a CCP for feed mixer problem, which is to select four materials to mix in order to design a cattle feed mix subject to protein and fat constraints, with the objective of minimizing the cost. That CCP problem, with four decision variables and four stochastic parameters, can be written as follows:

$$\max \quad 24.55x_1 + 26.75x_2 + 39.00x_3 + 40.50x_4$$

subject to mixed constraints:

$$\begin{aligned}
x_1 + x_2 + x_3 + x_4 &= 1, \\
2.3x_1 + 5.6x_2 + 11.1x_3 + 1.3x_4 &\geq 5, \\
P [\eta_1 x_1 + \eta_2 x_2 + \eta_3 x_3 + \eta_4 x_4 \geq 21] &\geq 0.8, \\
x_1, x_2, x_3, x_4 &\geq 0.
\end{aligned}$$

Where η_1, η_2, η_3 , and η_4 , have normal distributions $\mathcal{N}(12, 0.2809^2)$, $\mathcal{N}(11.9, 0.1936^2)$, $\mathcal{N}(41.8, 20.25^2)$, and $\mathcal{N}(52.1, 0.6241^2)$ respectively. A run of the genetic algorithm (200 generations in GA, 1000 simulations) shows that the optimal solution is $\mathbf{X}^* = (0.2, 0.35, 0.27, 0.17)$ with $f(\mathbf{X}^*) = 31.95$ **Problem VIII Insurance and Investment Portfolio Model**

The model described in section 5.5 was written in AMPL and submitted to NEOS Server Version 4.0. ACRS (AMPL input). The transformed objective $P(x \geq x_0)$ is defined in eqn 118. The model was solved with the GA developed in this chapter (MCS module can generate only independent random variables) assuming that the returns follow independent normal distributions, which is not the case in practice as they have skewed distributions. The results are given in table 1 for $\beta_1 = \beta_2 = 0.55$, $\alpha_1 = \alpha_2 = 0.30$.

Decision Variables	NEOS	
	ACRS(1.0.0)	GA
Allied lines	0.12	0.1640
Auto liability	0.276	0.5149
Auto physical damage	0.326	0.3088
Burglary	0.042	0.1840
Commercial multiple peril	0.12	0.142
Fire	0.375	0.3325
Homeowners	0.075	0.1659
Inland marine	0.41	0.0986
Ocean marine	0.33	0.1739
Worker's compensation	0.32	0.3154
Liquid asset	0.867	0.6557
Nonliquid asset	1.5	1.6030
obj = $P(x \geq x_0)$	0.67= $\Phi(0.4179)$	0.96

Table 1

CCP Results for Insurance and Investment Portfolio

5 Pension fund formulated as CCP: A Case study

We consider two pension fund models that have been formulated as SP problems for risk management. We study a prototype stochastic programming model for managing pension funds and a CCP model for insurance and investment portfolio management. SP models have been applied to financial planning problems by Casigli and Dempster (1998), Kusy and Ziemba (1986), Mulvey and Vladimirou (1992), Kouwenberg (2000), Dert(1995), Zenios(1995). The model of Dert(1995) includes chance constraints for solvency of pension funds. Vlerk [53] uses integrated chance constraints in the asset and liability management (ALM) model for pension funds. The implementation described in section 4.3 is demonstrated by developing a deterministic equivalent of the model [23] and solving the model.

5.1 Pension Fund Modelling Using Stochastic Programming

A pension fund is an institute that has been set the task to make benefit payments to people that have ended their career. The payments to be made to the retirees must be in accordance with the benefit formulae that prescribe

the flow of payments to which each participant in the fund is entitled. Contributions are, by definition, made by the sponsor of the fund. The sponsor can be the employer, the active participants, or both. At given points in time the value of the assets of the fund is increased by receiving contributions and by the appreciation of the value of invested assets and is decreased by making benefit payments. It is the responsibility of the pension fund to balance this process in such a way that the fund meets the solvency standards in force, and that all benefit payments, now and in the future, can be made timely. Consider a pension fund that conducts activities as follows: 1) collection of premiums from the sponsor and/or active employees; 2) investment of available funds; and 3) payment of pensions to the retired employees. Important decisions are 1)

levels of contributions have to be set in such a way that the sponsor of the fund is able and willing to pay them. The levels of contribution as percentage of the costs of wages have to be set. Annual hikes in contribution as percentage of the costs of wages, may not exceed a given level. 2) Allocation of assets over asset classes (**the asset mix**) in which the fund is willing to invest. There are upper and lower bounds on the percentage of assets to be invested in each category.

The level of future benefit payments is uncertain. It is subject to the development of the characteristics of the participants, which are determined future career paths, life and death etc. The major source of uncertainty is the future development of price inflation and wage inflation: at retirement, the level of old age pension is usually 70% of the final salary. This pension includes a state pension to a fixed amount. It follows that pension rights of active participants that have been earned over past years of service will be increased by wage inflation. The benefits of inactive participants are indexed with price inflation. Solvency requirements have been imposed to see that the pension fund has accumulated a level of assets that is sufficient to fund its liabilities. In order to finance the pension scheme, the plan sponsor each year pays a contribution to the fund. Next the pension fund has to decide how to invest these contributions, in order to meet the short term solvency requirements and to fulfill its long term obligations. The PF model therefore should include decisions about the investment strategy and the contribution payment strategy.

Multi-stage stochastic programming can be used to model the problem of pension fund management, given the stochastic data provided in the form of scenario trees. The mean or variance of the total wealth at the end of planning horizon can be optimized by either solving a linear stochastic program or a quadratic stochastic program, respectively. When the formulation is scenario based LP solvers can be used. When the model contains risk constraints, it becomes nonlinear and hence other solution techniques have to be used. A pension fund model with risk constraints is discussed in this chapter.

Issues in Pension Fund Management Some of the most important issues a pension fund manager has to face is the determination of optimal asset allocations over the time to product maturity are given below:

- (1) *Stochastic nature of the asset returns and liabilities*
The future asset returns and liability streams over the life of the product is unknown. liabilities, in particular, are determined by the actuarial events and have to be matched by the assets. Thus each allocation decision will have to take into account the liabilities level which, in turn, is directly linked to the contribution policy requested by the fund.
- (2) *Long investment horizons*
The typical investment horizon is very long (30 years). This means that the fund has to be rebalanced many times and can make Markowitz-style portfolio optimization inefficient. Various dynamic stochastic optimization techniques are needed to take explicitly into account the ongoing re-balancing of the asset-mix.
- (3) *Risk of under-funding*
There is a very important requirement to monitor and manage the probability of under-funding, that is the probability that the pension fund cannot meet its funding requirements.
- (4) *Management constraints*
The management of pension fund is dictated by a number of solvency constraints which are put in place by the appropriate regulating authorities.

Model Description

Variables, Coefficients and Parameters of the Model

Decision variables

$x_{i,t}$: money invested in asset i at time t

y_t : contribution rate at time t , i.e. premium paid by the sponsor and /or active employees as fraction of their wages at time t

Y_t - regular contribution over period t

Z_t : remedial contribution at time t

Stochastic variables

A_t : value of all assets owned by the fund at time t

W_t : wages earned by active members at time t

l_t : payments made by the fund to the retirees at time t

$r_{i,t}$: return on investment in asset i at period t

L_t : liabilities of the fund at time t

Model parameters

A_0 : total initial value of all assets

W_0 : total initial amount of all wages

λ - is a penalty parameter which reflects the preference of asking regular contributions over remedial contributions and its value should be so chosen that an optimal solution will not allow for remedial contributions in excess of the minimal amount required to restore solvency.

γ_t - discount factors reflect preferences with respect to the timing of contri-

bution payments. This is specified as a stochastic parameter rather than as a deterministic parameter. *Objectives and Constraints There are many possibilities for measuring the performance of the pension fund by an objective function. Natural candidates are 1) maximize the risk adjusted expected terminal wealth. 2) optimize an objective with a target level of return. Set the target return rate \hat{r} and penalize under achievement while encouraging earnings higher than \hat{r} . 3) minimize the expected cost of funding. In this model the objective is to minimize the expected cost of funding. The parameters in the objective function do not have a financial meaning necessarily, but one can change them in order to generate solutions from the model corresponding to various weights on different goals.

The objective function is

$$\text{minimize} \quad A_0 + Y_0 + \sum_{t=1}^{T-1} E[\gamma_t Y_t] + \lambda \sum_{t=1}^T E[\gamma_t Z_t] \quad (91)$$

There are upper and lower limits for the contribution rates

$$y_t^l \leq y_t \leq y_t^u \quad t = 0, \dots, T-1 \quad (92)$$

At each decision moment the following balance equation holds:

$$\sum_{i=0}^n x_{i,t} = A_t + W_t y_t - l_t \quad t = 0, \dots, T-1 \quad (93)$$

. The sum $\sum_{i=0}^n x_{i,t-1}$ invested at time $t-1$ results in the value of all assets at time t :

$$A_t = \sum_{i=0}^n x_{i,t-1}(1 + r_{i,t}) \quad t = 1, \dots, T \quad (94)$$

At time t , it should be decided to what percentage of the costs of wages the contribution for period t should be met, observing that it may not be raised by $100\beta_t$ percent of the cost of wages:

$$y_t - y_{t-1} \leq \beta_t \quad \forall t \quad (95)$$

At each time period, $t = 1, \dots, T$, we want with high certainty to satisfy the liability constraints:

$$A_t = \sum_{i=0}^n x_{i,t-1}(1 + r_{i,t}) \geq L_t \quad (96)$$

The ratio of assets to liabilities, A_t/L_t is usually referred to as the *funding ratio* of the pension fund. A target funding ratio of ψ can easily be incorporated by replacing the above constraint by

$$A_t = \sum_{i=0}^n x_{i,t-1}(1 + r_{i,t}) \geq \psi L_t \quad \forall t \quad (97)$$

with high certainty. Values of $\psi > 1$ often are used to add some extra safety margin to the constraint. It is often impossible that the constraint 97 will be satisfied with probability 1.0. Therefore, this constraint is relaxed, and we would like to find a solution with a sufficiently high probability of meeting the liability constraints while keeping costs at a reasonable level.

Chance Constraint

The solvency constraint can be included in a natural and consistent way by formulating it as a chance constraint:

$$Pr [A_t < \psi L_t] \leq \alpha \tag{98}$$

VaR Constraint

When equation 97 is violated we have a loss and the pension fund is underfunded. As measure of this loss, we use the difference between the right-hand side and left-hand side in 97:

$$f_\psi(x, r, L) = \psi L - \sum_{i=0}^n (1 + r_{i,t})x_i \tag{99}$$

Hence, 99 could be replaced with high certainty.

Let p be the joint probability measure of the vector (r, L) and denote by $\Phi_\psi(\zeta, x)$ the cumulative probability distribution of the loss, given x ; by equation 53

$$\Phi_\psi(\zeta, x) = P(f_\psi(x, r, L) \leq \zeta) \tag{100}$$

$$= \int_{f_\psi(x,r,L) \leq \zeta} p(dr, dL) \tag{101}$$

which is by definition is the probability that the loss $f_\psi(x, r, L)$ do not exceed a threshold value ζ . Now if α is the confidence interval that 97 is not violated, the inequality in 97 can be expressed as follows:

$$\zeta_{\alpha,\psi}(x) \leq 0 \tag{102}$$

where

$$\zeta_{\alpha,\psi}(x) = \min \{ \zeta \in R : \Phi_\psi(x, \zeta) \geq \alpha \}$$

The value $\zeta_{\alpha,\psi}(x)$ is called the α -VaR as given in section 2.3 and the constraint 102 means that the loss in at least $100\alpha\%$ of outcomes must be below or equal to 0 (note that, in general, this threshold level may be chosen to be different from 0).

CVaR Constraint

CVaR is the weighted average of VaR and the losses exceeding VaR (denote it by $CVaR_\alpha(x)^+$ and assuming there are losses strictly exceeding VaR). Using 56, CVaR is then defined as follows:

$$CVaR_\alpha(x) = \lambda VaR_\alpha(x) + (1 - \lambda) CVaR_\alpha(x)^+ \tag{103}$$

where the weight equals

$$\lambda = [\Phi(x, VaR_\alpha(x)) - \alpha] / [1 - \alpha] \in [0, 1]$$

(see 53). Note that, when the distribution of the losses has continuous density, $\lambda = 0$, and we have $CVaR_\alpha(x) = CVaR_\alpha(x)^+$. This is not the case when the distribution is discrete (or is approximated by a discrete distribution using sampling of scenarios). CVaR is convex, which makes it possible to construct efficient algorithms for controlling CVaR. Since CVaR always exceeds or equals VaR, we could replace 102 by

$$CVaR_\alpha(x) \leq w \tag{104}$$

for some w . With $w = 0$, we have a risk constraint that dominates the VaR constraint 102. This CVaR constraint can be transformed into a system of linear constraints [52,22].

5.2 Scenario Generation

The probability distribution P of the random parameters is an important input to the model. Discretization of the continuous distribution results in a description of the stochastic elements in the form of a scenario tree. This is called scenario generation. The goal of scenario generation is to get realizations for the uncertain parameters in each node of the scenario tree. The solution found by the optimization algorithm heavily depends on the stochastic scenario model of the future. These realizations have to satisfy certain criteria, as explained below.

- They should be consistent with the historical data, including any interaction amongst the variables.
- They should not exhibit arbitrage opportunities

In order to fulfil the first requirement, the variables are modelled according to Vector Autoregressive (VAR) Model [9]. If we use methods such as time series analysis or parameters implied by the current market condition, the scenario generation procedure consists of the following steps:

- Gather the financial time series and macroeconomic data and/or the current market information
- Run the forecasting method (econometric model and/or implied parameters) on this data. This gives the forecasted probability distribution, e.g. the forecasted mean and variance
- Use the Brownian Bridge to simulate rate paths up to the forecast horizon while varying the final interest rate.

A detailed description of VAR scenario generation can be found in [5]. The scenario tree generated using the above technique is relatively simple (due to coarse discretization), but leads to very large optimization models. Scenario reduction/aggregation techniques [13,30,9] have been used to reduce the dimension of the problem.

The Stochastic Optimization System In dynamic stochastic optimization, the unfolding of stochastic variables is represented by a large number of scenarios and decisions are made in stages according to the tree representations of the future data. At each stage, decisions are made based on the information revealed up to that point, so the decision variables at a stage are functions of the random variables revealed up to that point. The reader is referred to [6,9] for discretization of the conceptual SP model for pension funds.

Figure 5 sketches the structure of the overall optimization system. The sce-

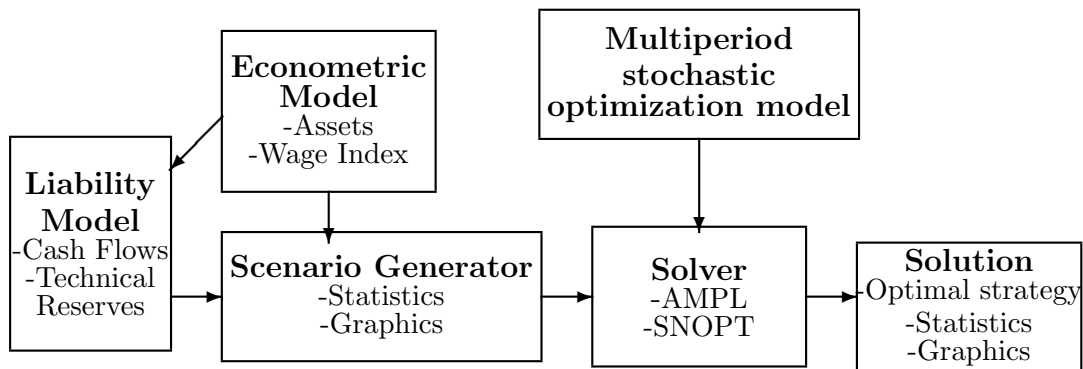


Fig. 5. Stochastic optimization system for pension fund management

nario generator takes as inputs, the period and branching structures of the scenario tree and the time series model for the stochastic factors and generates the scenario tree for the assets and liabilities. The scenario tree is written into a text file in AMPL format described in Fourer [38,39]. The optimization model written in AMPL modelling language and the data from the scenario generator are processed in AMPL to generate the MPS file, fed to a solver (SNOPT or MINOS).

5.3 Insurance and Investment Portfolio Model Using CCP

Li [23] has developed an insurance and investment portfolio model that was formulated in terms of \mathbf{P} -models of CCP. The model is to maximize the probability of achieving the aspiration level under two chance constraints and other regulatory and institutional constraints. The first chance constraint specifies

that the probability that the total return will fall below the insurer's acceptable minimum level will not exceed their return risk level. Second chance constraint allows the insurer's minimum requirement on cash and liquid assets to be violated, but not more than their liquidity risk level. In this portfolio model, the rates of return can follow skewed distributions different from the traditional mean-variance portfolio models where the rates of returns are assumed to follow normal distributions. The objectives in the traditional portfolio literature are to maximize expected returns or to minimize the variance of returns subject to budget requirements and/or minimum returns on equity. In contrast the objective of this model is to maximize the insurer's probability of obtaining at least his/her aspiration level of return on equity subject to two chance constraints that are used to achieve and maintain the company's minimum return requirement and liquidity.

5.4 The Model

Indices

m the number of types of possible investments

n the number of candidate insurance lines

a_i ratio of investment in asset i to equity ($i = 1, 2, 3, \dots, m$)

or ratio of insurance premium for line i to equity ($i = m + 1, \dots, m + n$)

r_i rate of return on investment ($i = 1, 2, 3, \dots, m$) or rate of profit on the insurance activity ($i = m + 1, \dots, m + n$)

x the return on equity

δ an insurance leverage

y cash and liquid assets

g_i the funds-generating factor

d_i the rate of cash returns on investment $k + 1 \leq i \leq m$

l_i the liquid part of the i -th asset, $k + 1 \leq i \leq m$

k the number of liquid assets

Suppose the dollar incomes from investment and insurance activities are random variables with known distributions having finite expected values and variances. For any given period, the return on equity, x is equal to profit plus investment income i.e.

$$x = \sum_{i=1}^m a_i r_i + \sum_{i=m+1}^{m+n} a_i r_i \quad (105)$$

the average value and the variance of return on equity are

$$E(x) = \sum_{i=1}^{m+n} a_i E(r_i), \quad Var(x) = \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} a_i a_j Cov(r_i, r_j) \quad (106)$$

The following constraints are imposed:

1. maximum and minimum limits on premium allocations on to each line of insurance.

$$a_i^{min} \leq a_i \leq a_i^{max}, \quad (i = m + 1, \dots, m + n) \quad (107)$$

2. constraints on the maximum and minimum amounts to be invested in each of the different kinds of investments:

$$a_i^{min} \leq a_i \leq a_i^{max}, \quad (i = 1, \dots, m) \quad (108)$$

Since the insurance industry is required to maintain a balance between the sources of investable funds and their uses, we have the following balance sheet constraint:

total assets per one dollar of equity = 1 + total liabilities which are generated by premiums

$$\sum_{i=1}^m a_i = 1 + \sum_{i=m+1}^{m+n} a_i g_i \quad (109)$$

the parameter g_i is called the funds-generating factor which is the ratio of liabilities to premium for insurance line i . Regulation, marketing constraints, and various other rules of thumb used in the insurance industry usually limit the amount of premium volume for a given amount of equity. This is expressed by the following equation:

$$\sum_{i=m+1}^{m+n} a_i = \delta \quad (110)$$

where the parameter δ reflects how much premium is written for each dollar. Insurers must have enough cash and liquid assets, y , to meet cash requirements. The cash and liquid aspects of equity are represented by the return on insurance portfolio

$$\sum_{i=m+1}^{m+n} a_i r_i \quad (111)$$

liquid assets and return on liquid assets

$$\sum_{i=1}^k a_i (1 + r_i) \quad (112)$$

interest or other cash income from liquid parts of the non-liquid assets

$$\sum_{i=k+1}^m a_i (l_i + d_i) \quad (113)$$

Hence, we have

$$y = \sum_{i=1}^k a_i (1 + r_i) + \sum_{i=k+1}^m a_i (l_i + d_i) + \sum_{i=m+1}^{m+n} a_i r_i \quad (114)$$

and

$$lE(y) = \sum_{i=1}^k a_i E(r_i) + \sum_{i=k+1}^m a_i (l_i + E(r_i)) + \sum_{i=m+1}^{m+n} a_i E(r_i) \quad (115)$$

$$Var(y) = \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} a_i a_j Cov(r_i, r_j) = Var(x) \quad (116)$$

Define the insurer's return risk as the probability that their return on equity is less than or equal to their minimum level β_1 and the insurer's liquidity risk as the probability that their cash and liquid assets are less than or equal to their minimum requirement β_2

When the portfolio selection is determined, the ratios of all assets and insurance premiums to equity are to be constrained by the following probabilistic constraints:

3. $P(x \leq \beta_1) \leq \alpha_1$, α_1 maximum probability that the insurer's return on equity will fall below the minimum risk level β_1 .

4. $P(x \leq \beta_2) \leq \alpha_2$, α_2 maximum probability that the insurer's cash and liquid assets will fall below the minimum risk level β_2

Objective

Let the insurer's aspiration level be x_0 , for their return on equity. The insurer's objective is, for given risk levels and aspiration level, to maximize the probability of achieving this aspiration level, subject to several chance and deterministic constraints. In more precise form the insurer's objective is

$$\max_{a_i} P(x \geq x_0) \quad (117)$$

subject to the above four constraints.

5.5 The Deterministic Model

We can make use of the transformation given in Case(III) of section 2.2. Thus the above CCP model was transformed to the following deterministic version for solving using NLP solvers.

$$\max_{a_i} ((E(x) - x_0)) / (Var(x))^{1/2} \quad (118)$$

subject to

$$\begin{aligned}
E(x) + \Phi(\alpha_1) (Var(x))^{1/2} &\geq \beta_1, \\
E(y) + \Phi(\alpha_2) (Var(y))^{1/2} &\geq \beta_2, \\
\sum_{i=1}^m a_i &= 1 + \sum_{i=m+1}^{m+n} a_i g_i, \\
\sum_{i=m+1}^{m+n} a_i &= \delta, \\
a_i^{min} &\leq a_i \leq a_i^{min}, \quad (i = m + 1, \dots, m + n)
\end{aligned} \tag{119}$$

where

$$\begin{aligned}
x &= \sum_{i=1}^m a_i r_i + \sum_{i=m+1}^{m+n} a_i r_i \\
y &= \sum_{i=1}^k a_i (1 + r_i) + \sum_{i=k+1}^m a_i (l_i + d_i) + \sum_{i=m+1}^{m+n} a_i r_i
\end{aligned}$$

6 Conclusion

The use of GAs in stochastic programming has been limited. We developed a GA and Monte Carlo for chance constrained programming. The population of the GA was evaluated using a fitness function that was created with a multiplicative penalty function. The penalty function was made with the product of degree of satisfaction of the constraints. Monte Carlo simulation was used for evaluating the stochastic functions. The advantage of GA is that it can be combined with other techniques to obtain problem-specific solution algorithms. Also, it can solve highly nonlinear, non-convex optimization problems without finding the first or second derivatives of the functions.

Contrary to the linear case, the integer and non-linear cases have received limited attention in the literature. It appears unlikely that general-purpose algorithms will solve such problems exactly. Instead, we anticipate the development problem specific approximation schemes for integer and nonlinear problems. There are several opportunities for the development and application of global optimization algorithms for stochastic programming. We refer the reader to the thesis of Bastin [4] for a more detailed discussion of algorithms for nonlinear stochastic programming

Most practical and theoretical results assumes that the program to be solved is convex. A lot of advances in nonlinear non-convex deterministic programming have been achieved in recent years. Powerful algorithms are known (see *Numerical Optimization* by Nocedal and Wright [32]), but efforts to transform them in the stochastic case have been limited. Nonlinear applications and algorithms both convex and non-convex, are often multi-stage problems with scenario formulations. They can be written as nonlinear programs with

the size growing with the number of scenarios. The challenge consists then to exploit the structure of problem while solving it in limited amount of time. Also procedures are usually parallelizable, so that they can benefit from the development of multiprocessing computing. Parallel methods attract many researchers both in linear and nonlinear stochastic optimization.

7 Future Work

Future direction of research should investigate new strategies for improving the performance. Research can be focused on developing a hybrid form of algorithms, which combines the good features of different algorithms like GA, tabu search and simulated annealing. Use of high quality genetic operators which maintains the diversity of the population and exploit the convexity and smoothness in the problem formulation should be explored. Design better evaluation function for evaluating the fitness of solutions. In this work we examined the use of GAs in stochastic programming when the decision variables are anticipative. Many risk management problems in finance are modelled as multi-period recourse problems. Further research can focus on developing combined algorithms for improving speed efficiency.

The disadvantage of using GA is that the optimal solution is not guaranteed within a finite computing time. For increased efficiency, the population in GA could be easily evaluated in parallel computations.

The following aspects in SP would set the pace for future trends:

- (1) The advances in computing techniques and in the algorithms for NLP are likely to have a great impact on the various models of stochastic programming; in particular, the central common theme between the various approaches would appear to be more and more clear. This trend is likely to continue from static to dynamic models of stochastic programming.
- (2) The sequential basis of decision-making in many practical situations would lead to the emphasis on the robustness of solutions and the Bayesian methods of suitable revision of prior estimates. The whole field of non-parametric estimation and decision-making would have its impact on the development of SP. Recent developments in sequential procedures of estimation applied to estimation problems are viewing estimation as a part of the decision-making process and this outlook is expected to have a significant impact on the decision-theoretic implications of SP.
- (3) The method of simulated optimization, which provide an insight into the implications of several plausible probability distributions, are going to have considerable influence in the development of SP. Already in stochastic control and statistic programming problems over time, simulated optimization methods found some acceptance and it is hoped similar methods would be extended to static and other areas of stochastic programming.

References

- [1] A. Charnes, W. C.: 1959, 'Chance constrained programming'. *Management Science* **6**, 73–79.
- [2] Albrecht, P.: 2004, 'Risk Measures'. *Encyclopedia of Actuarial Science*, John Wiley and Sons.
- [3] Barbosa, H. J. C. and A. C. C. Lemonge: 2003, 'A new adaptive penalty scheme for genetic algorithms'. *Information Sciences* **156**(3-4), 215–251.
- [4] Bastin, F.: 2001, 'Nonlinear stochastic programming'. Mmoires, FUNDP. Facult des sciences. Dpartement de mathmatique.
- [5] Berkelaar, A., H. Hoek, and A. Lucas: 1999, 'Arbitrage and Sampling Uncertainty in Financial Stochastic Programming Models'. Technical report, Erasmus University Rotterdam, Econometric Institute.
- [6] Bogentoft E., R. H. and S. Uryasev: 2001, 'Asset/Liability Management for Pension Funds Using CVaR Constraints'. *The Journal of Risk Finance* **3**(1), 57–71.
- [7] Costa, L. and P. Oliveira: 2001, 'Evolutionary Algorithms Approach to the Solution of Mixed Integer Non-Linear Programming Problems'. *Computers & Chemical Engineering* **25**, 257–266.
- [8] Crossley, W. A. and E. A. Williams: 1997, 'A Study of Adaptive Penalty Functions for Constrained Genetic Algorithm Based Optimization'. In: *AIAA 35th Aerospace Sciences Meeting and Exhibit*. Reno, Nevada.
- [9] Dert, C.: 1995, 'Asset liability management for pension funds, a multistage chance constrained programming approach'. Ph.D. thesis, Erasmus University, Rotterdam, The Netherlands.
- [10] Dowd, K.: 1998, *Beyond Value at Risk: The New Science of Risk Management*. Wiley, 1 edition.
- [11] F.Andersson, H.Mausser, D. and S.Uryasev: 2001, 'Credit Risk Optimization with Conditional Value-At-Risk Criterion'. *Mathematical Programming* (Series B 89), 273–291.
- [12] Goldberg, D.: 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York.
- [13] H. Heitsch, W. R.: 2003, 'Scenario reduction algorithms in stochastic programming'. *Computational Optimization and Applications* **24**, 187–206.
- [14] Homem-De-Mello, T.: 2003, 'Variable-sample methods for stochastic optimization'. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **13**(2), 108–133.

- [15] Ierapetritou, M.G., J. A. and E. Pistikopoulos: 1996, 'An Optimization Approach for Process Engineering Problems under Uncertainty'. *Computers and Chemical Engineering* **20**, 703–709.
- [16] Inuiguchi, M. and J. Ramik: 2000, 'Probabilistic linear programming: a brief review of fuzzy mathematical programming and a comparison with stochastic programming in portfolio selection problem'. *Fuzzy Sets and Systems* **111**, 3–28.
- [17] Janne Makinen, K. M. and M. M. Makela: 1999, 'SOME PENALTY METHODS WITH GENETIC ALGORITHMS'. P.O. Box 35 (MaE), FIN-40351 Jyva”skyla”, FINLAND.
- [18] Jiafu Tang, Dingwei Wang, A. I. and R. Y. K. Fung: 1998, 'A hybrid genetic algorithm for a type of nonlinear programming problem'. *Computers & Mathematics with Applications* **36**(5), 11–21.
- [19] Kaifeng Hou, Yourun Li, J. S. and S. Hu: 2000, 'Delayed sampling approach to stochastic programming in chemical processes'. *Computers and Chemical Engineering* **24**, 619–624.
- [20] Kall, P. and S. W. Wallace: 1994, *Stochastic programming*. New York, NY: Wiley, (ISBN 0-471-95108-0 /hc). xii, 307 p. sterling 49.95.
- [21] Kampas, A. and B. White: 2003, 'Probabilistic programming for nitrate pollution control: Comparing different probabilistic constraint approximations'. *European Journal of Operational Research* **147**(1), 217–228.
- [22] Krokmal, P., J. and S. Uryasev: 2002, 'Portfolio Optimization with Conditional Value-At-Risk Objective and Constraints'. *The Journal of Risk* **4**(2).
- [23] Li, S. X.: 1995, 'An Insurance and Investment Portfolio Model Using Chance Constrained Programming'. *Omega: International Journal of Management Science* **23**, 577–585.
- [24] Li, Y.-X. and M. Gen: 1996, 'Nonlinear mixed integer programming problems using genetic algorithm and penalty function'. In: *Proceedings of the 1996 IEEE International Conference on Systems, Man and Cybernetics*, Vol. 4. pp. 2677–2682.
- [25] Liu, B. and K. Iwamura: 1996, 'A Genetic algorithm for chance constrained programming'. *Journal Information and Optimization sciences* **17**(2), 409–422.
- [26] M. P. Biswal, N. P. B. and D. Li: 1998, 'Probabilistic linear programming problems with exponential random variables: A technical note'. *European Journal of Operational Research* **111**(3), 589–597.
- [27] M.Gen and R.Chen: 1996, *Genetic Algorithms and Engineering Design*. Wiley, New York.
- [28] Michalewicz: 1995, 'A survey of constraint handling techniques in Evolutionary Computation methods'. *Evolutionary programming IV* pp. 135–155.

- [29] Michalewicz, Z.: 1994, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 2 edition.
- [30] N. Grwe-Kuska, H. Heitsch, W. R.: 2003, ‘Scenario reduction and scenario tree construction for power management problems’. In: M. P. A. Borghetti, C.A. Nucci (ed.): *IEEE Bologna Power Tech Proceedings*.
- [31] Nanakorn, P. and K. Meesomklin: 2001, ‘An adaptive penalty function in genetic algorithms for structural design optimization’. *Computers and Structures* **79**(29-30), 2527–2539.
- [32] Nocedal, J. and S. J. Wright: 1999, *Numerical Optimization*, Operations Research. Springer Verlag.
- [33] P. Fsek, P. Kall, J. M. S. S. and S. Siegrist: 2000, ‘Multistage stochastic linear programming: Aggregation, approximation, and some open problems’. Technical report, Oper. Res., University of Zurich.
- [34] Panne VD, P. V.: 1963, ‘Minimum-Cost Cattle Feed under Probabilistic Protein Constraints’. *Management Science* **9**(3), 405–430.
- [35] Pistikopoulos, E. and M. Ierapetritou: 1995, ‘A Novel Approach for Optimal Process Design under Uncertainty’. *Computers and Chemical Engineering* **19**, 1089–1110.
- [36] Prekopa, A.: 1993, ‘Programming under probabilistic constraint and maximizing a probability under constraints’. Rrr 35-93, Center for Operations Research, Rutgers university, P.O. box 5062, New Brunswick, New Jersey.
- [37] Prekopa, A. and O. Fiedler: 1995, ‘Bounding in Multi-Stage Stochastic Programming Problems’. Rrr 24-95, Center for Operations Research, Rutgers university, P.O. box 5062, New Brunswick, New Jersey.
- [38] R. Fourer, D. G. and B. W. Kernighan: 1993, *AMPL: A Modeling Language for Mathematical Programming*. The Scientific Press (now part of Wadsworth Publishing).
- [39] R. Fourer, D. M. G. and B. W. Kernighan: 1994, ‘An Introduction to the AMPL Modeling Language for Mathematical Programming’. *Mathematech* **1**(1), 49–56.
- [40] Richardson, J.T., M. P. G. L. and M. Hilliard: 1989, ‘Some Guidelines for Genetic Algorithms with Penalty Functions’. In: *Proc. Of the 3th Inter. Conf. On Genetic lgorithms*. Los Altos, CA, pp. 191–197.
- [41] Rockafellar, R. and S. Uryasev.: 2000, ‘Optimization of Conditional Value-At-Risk’. *The Journal of Risk* **2**(3), 21–41.
- [42] Rubinstein, R.: 1981, *Simulation and Monte Carlo Method*. John Wiley and Sons.
- [43] S. B. Sinha, S. H. and M. P. Biswal: 2000, ‘Fuzzy programming approach to multi-objective stochastic programming problems , when b_i ’s follow joint normal distribution’. *Fuzzy Sets and Systems* **109**(1), 91–96.

- [44] Salah E. Elmaghraby, H. S. and M.-J. Yao: 2001, ‘Chance-constrained programming in activity networks: A critical evaluation’. *European Journal of Operational Research* **131**(1), 440–458.
- [45] Shapiro, A.: 1991, ‘Asymptotic analysis of stochastic programs’. *Ann. Oper. Res.* **30**, 169-186.
- [46] Shapiro, A.: 1993, ‘Asymptotic behavior of optimal solutions in stochastic programming’. *Math. Oper. Res.* **18**, No.4, 829-845.
- [47] Siddharth Alexander, T. F. C. and Y. Li: 2003, ‘Minimizing CVaR and VaR for a Portfolio of Derivatives’. In: *International Conference on Modeling, optimization and Risk Management in Finance*. Cornell Theory Center, Cornell University, NY.
- [48] Smith, J.: 1999, ‘Optimizing Platform Survivability Using a Chance Constrained Linear Program’. In: *US Army Ground Vehicle Survivability Symposium*. Survivability/lethality analysis directorate, munitions and Platform branch, White sands Missile range, NM.
- [49] S.P.Simonovic: 2002, ‘Understanding Risk Management’. In: *Annual Conference of the Canadian Society for Civil Engineering*. Montral, Qubec, p. 12pp.
- [50] Suwarna Hulsurkar, M. B. and S. Sinha: 1997, ‘Fuzzy programming approach to multi-objective stochastic linear programming problems’. *Fuzzy Sets and Systems* **88**, 173–181.
- [51] Takao Yokota, M. G. and Y. xiu Li: 1996, ‘Genetic algorithms for non-linear mixed integer programming problems and its applications’. *Computers and Industrial Engineering* **30**(4), 905–907.
- [52] Uryasev, S.: 2000, ‘Conditional Value-at-Risk: Optimization Algorithms and Applications’. *Financial Engineering News* (14), 1–5.
- [53] Vlerk, M. v. d.: 2003, ‘Integrated chance constraints in an ALM model for pension funds’. Technical Report 03A21, University of Groningen, Research Institute SOM (Systems, Organisations and Management).
- [54] Y. Tsompanakis, M. P.: 2000, ‘Robust and efficient methods for reliability-based structural optimization’. In: *Fourth International Colloquium on Computation of Shell and Spatial Structures*. Chania - Crete, Greece.