# CHAPTER 4

# STOCHASTIC SIMULATION BASED GENETIC ALGORITHM FOR CHANCE CONSTRAINED FRACTIONAL PROGRAMMING PROBLEMS

## 4.1 INTRODUCTION

Addressing data uncertainties in mathematical programming models has been a central problem in optimization since a long. One of the methods that have been proposed to address data uncertainty over the years is Stochastic Programming (SP). Stochastic programs are mathematical programs, where some of the data incorporated in the objective function or constraints are uncertain. Uncertainty is usually characterised by a probability distribution on the parameters. In real life situations, there are many applications of SP. Some of these can be seen in Charnes and Cooper (1954) and Jeeva et al (2002, 2004). Two major approaches to stochastic programming (Goicoechea et al 1982; Kambo 1984) are recognized as

(i) Chance Constrained Programming (CCP)

(ii) Two stage programming.

The CCP technique is one which can be used to solve problems involving chance constraints i.e., constraints having finite probability of being violated. The CCP was originally developed by Charnes and Cooper (1959). In recent years, it has been generalized in several directions and has various applications.

Fractional Programming (FP) is an optimization problem in which ratio of two linear functions is optimized subject to some constraints (Charnes and Cooper 1962; Zionts 1968). Basic concepts about Chance Constrained Fractional Programming (CCFP) are available in Charles et al (2001), Charles and Dutta (2003). In contrast to the unifying role of the Simplex method in Linear Programming, there exists no unique approach to chance constrained optimization. Many approaches have been proposed to solve the CCP problems. The most familiar approach adopted by many researchers so far has been deriving deterministic equivalent of stochastic objective function and/or chance constraints of CCFP problems. Generally deriving the deterministic equivalence is very difficult due to complicated multivariate integration and is possible if the random variables involved in the stochastic objective function and/or chance constraints follow some specific distributions such as Normal, Uniform, Exponential and Lognormal distributions. Therefore this approach combines parametric approach with stochastic simulation based genetic algorithm for solving CCFP problems, where random variables can follow any continuous distribution. The advantage of the proposed approach is that deriving deterministic equivalence is not required. Here chance constraints are directly used within the genetic process and their feasibilities are checked by the stochastic simulation technique.

## 4.2 MOTIVATION

Finding a solution to CCFP using conventional approach such as the conversion of stochastic objective function and/or constraints to their respective deterministic equivalents requires more computational effort. This is because the converting process is usually hard and only successful for some special cases. Moreover after the conversion, the resultant deterministic model has to be solved by another traditional procedure. Motivated by this, a new approach has been developed which avoids more computational effort.

With the development of modern computers, complex CCP models without deterministic equivalents have been solved by innovative computations, for example, stochastic simulation based genetic algorithms.

## 4.3 PROBLEM FORMULATION

CCFP offers a way to deal with planning in a situation where the problem parameters are not known with certainty. Such situations arise, where technological aspects of the system under study may be highly complicated to observe fully.

The general form of CCFP problem is given below

$$\text{Max } R(X,\xi) = \frac{N(X,\xi)+\gamma}{D(X,\xi)+\beta} \tag{4.1}$$

$$\text{subject to } \Pr\left[g_i(X,\xi) \le 0\right] \ge \alpha_i, \qquad i=1, 2,\ldots, m \tag{4.2}$$

$$0 \le x_j \le x_j^{ul}, \qquad j=1, 2,\ldots, n \tag{4.3}$$

$$\alpha_i \in (0,1), \qquad i=1, 2,\ldots, m \tag{4.4}$$

where $X = \{x_1, x_2,\ldots,x_n\}$ is the decision vector, $X \in \mathbb{R}^n$, $X^{ul} = \left\| x_j^{ul} \right\|$, $x_j^{ul}$ is the upper limit value of the $j^{th}$ ($j=1, 2,\ldots, n$) decision variable, $\xi = \{\xi_1, \xi_2,\ldots,\xi_t\}$ is the stochastic vector , $\xi \in \mathbb{R}^t$ , which follows some continuous probability distribution with known parameters. In this model, $N(X,\xi)$, $D(X,\xi)$ and $g_i(X,\xi)$ are real valued functions. $N(X,\xi)$: $\mathbb{R}^{n+t} \to \mathbb{R}$ and $D(X,\xi)$: $\mathbb{R}^{n+t} \to \mathbb{R}$ are the numerator and denominator of the objective function $R(X,\xi)$: $\mathbb{R}^{n+t} \to \mathbb{R}$ . $g_i(X,\xi)$: $\mathbb{R}^{n+t} \to \mathbb{R}$ are the stochastic constraints. $\alpha_i$ is the minimum probability measure that the chance constraints sets are required to satisfy and $\gamma, \beta$ are scalars.

$$\text{Let } S = \left\{ X / \Pr\left[ g_i(X,\xi) \le 0 \right] \ge \alpha_i, 0 \le X \le X^{ul}, X \in \mathbb{R}^n \right\}.$$

CCFP problem is one of optimization problems, which can be solved by different techniques within the constraint satisfaction paradigm. The problem here is to develop an algorithm for solving CCFP problem.

## 4.4 STOCHASTIC PARAMETRIC MODEL FOR CCP PROBLEM WITH FRACTIONAL OBJECTIVE

Parametric method is widely used method to solve Linear and Non-linear FP problems when compared to Charnes and Cooper (1962) transformation, which is used only for solving Linear FP problems. This section systematically develops the required background to support the CCFP model in accordance with GA. The parametric model for CCFP can be defined in two ways namely E-model and V-model. The E-model attempts to maximize the expected value of the objective function whereas the V-model attempts to minimize variance of the objective function. The prime focus is now on E-model CCFP.

A well known parametric method for solving FP can be seen in Pardalos and Phillips (1991) and is defined as $\underset{X \in S}{Max}\left[ (N(X) + \gamma) - \lambda (D(X) + \beta) \right]$, where $\lambda \in \mathbb{R}$ is a constant. $\lambda^0, \lambda^1, \lambda^2, \ldots$ generated by the algorithm (Dinkelbach 1967), is strictly monotone increasing. However, Dinkelbach's method is defined for a deterministic model, whereas in a probabilistic model $N(X), D(X)$ and $\lambda$ are stochastic in nature. The parametric forms of E-model and V-model for CCFP problem are defined as given below.

Parametric form of E-model: a function that maximizes the expected value of the objective function

$$\Lambda^{E_\xi}(X,\lambda) = \text{Max}\left[(N^{E_\xi}(X)+\gamma) - \lambda(D^{E_\xi}(X)+\beta)\right] \tag{4.5}$$

satisfying the probabilistic constraints given in system (4.1), where $N^{E_\xi}(X) = E_\xi(N(X,\xi))$, $D^{E_\xi}(X) = E_\xi(D(X,\xi))$ and $E_\xi[.]$ denote the expectation operator with respect to $\xi$.

Parametric form of V-model: a function that minimizes the variance value of the objective function

$$\Lambda^{V_\xi}(X,\lambda) = \text{Min}\left[N^{V_\xi}(X) - \lambda^2 D^{V_\xi}(X)\right] \tag{4.6}$$

satisfying the probabilistic constraints given in system (4.1), where $N^{V_\xi}(X) = V_\xi(N(X,\xi))$, $D^{V_\xi}(X) = V_\xi(D(X,\xi))$ and $V_\xi[.]$ denote the variance operator with respect to $\xi$.

**Theorem 4.1:** If $R(X,\xi)$ is a convex function, then so is the function

$$\Lambda^{E_\xi}(X,\lambda) = \underset{X\in S}{Max}\left[(N^{E_\xi}(X)+\gamma) - \lambda(D^{E_\xi}(X)+\beta)\right]$$

**Proof.** Let $(X,\lambda) = \delta(x_1,\lambda_1)+(1-\delta)(x_2,\lambda_2)$ for $\lambda_1 \in \mathbb{R}$, $\lambda_2 \in \mathbb{R}$, $\delta \in (0,1)$, $x_1, x_2 \in \mathbb{R}^n$. It is given that $R(X,\xi)$ is a convex function. Then one has

$$\Lambda^{E_\xi}(X,\lambda) = \Lambda^{E_\xi}\left[\delta(x_1,\lambda_1)+(1-\delta)(x_2,\lambda_2)\right]$$

$$= \underset{X\in S}{Max}\left\{N^{E_\xi}(X)+\gamma - \left[\delta(x_1,\lambda_1)+(1-\delta)(x_2,\lambda_2)\right](D^{E_\xi}(X)+\beta)\right\}$$

$$=\underset{X\in S}{Max}\left\{N^{E_\xi}(X)+\gamma+\left[-\delta\ \ \delta\right](N^{E_\xi}(X)+\gamma)-\left[\delta(x_1,\lambda_1)+(1-\delta)(x_2,\lambda_2)\right]\right.$$

$$\left.(D^{E_\xi}(X)+\beta)\right\}$$

$$=\underset{X\in S}{Max}\left\{N^{E_\xi}(X)+\gamma-(x_2,\lambda_2)(D^{E_\xi}(X)+\beta)+\right.$$

$$+\delta\left[N^{E_\xi}(X)+\gamma-(x_1,\lambda_1)(D^{E_\xi}(X)+\beta)\right]$$

$$\left.-\delta\left[N^{E_\xi}(X)+\gamma-(x_2,\lambda_2)(D^{E_\xi}(X)+\beta)\right]\right\}$$

$$=\underset{X\in S}{Max}\left\{\delta\left[N^{E_\xi}(X)+\gamma-(x_1,\lambda_1)(D^{E_\xi}(X)+\beta)\right]\right.$$

$$\left.+(1-\delta)\left[N^{E_\xi}(X)+\gamma-(x_2,\lambda_2)(D^{E_\xi}(X)+\beta)\right]\right\}$$

$$\leq\delta\underset{X\in S}{Max}\left[N^{E_\xi}(X)+\gamma-(x_1,\lambda_1)(D^{E_\xi}(X)+\beta)\right]$$

$$\left.+(1-\delta)\underset{X\in S}{Max}\left[N^{E_\xi}(X)+\gamma-(x_2,\lambda_2)(D^{E_\xi}(X)+\beta)\right]\right\}$$

$$=\delta\ \Lambda^{E_\xi}\ (x_1,\lambda_1)\ +(1-\delta)\ \Lambda^{E_\xi}\ (x_2,\lambda_2)$$

It is proved that $\Lambda^{E_\xi}(X,\lambda)=\delta\ \Lambda^{E_\xi}\ (x_1,\lambda_1)\ +(1-\delta)\ \Lambda^{E_\xi}\ (x_2,\lambda_2)$.

Hence $\Lambda^{E_\xi}(X,\lambda)$ is convex function. □

**Property 4.1:** $\Lambda^{E_\xi}(X,\lambda)=\underset{X\in S}{Max}\left[(N^{E_\xi}(X)+\gamma)-\lambda(D^{E_\xi}(X)+\beta)\right]$ is continuous.

**Proof:** Let $\varepsilon>0$ be given. Now, check the continuity of $\Lambda^{E_\xi}(X,\lambda)$ at $\lambda_0\in\mathbb{R}$ being arbitrary.

$$\left|\Lambda^{E_\xi}(X,\lambda)-\Lambda^{E_\xi}(X,\lambda_0)\right| = \left|\underset{X\in S}{Max}\left\{N^{E_\xi}(X)+\gamma-\lambda(D^{E_\xi}(X)+\beta)\right\}\right.$$

$$\left. -\underset{X\in S}{Max}\left\{N^{E_\xi}(X)+\gamma-\lambda_0(D^{E_\xi}(X)+\beta)\right\}\right|$$

$$\leq\left|\underset{X\in S}{Max}\left\{\left\{N^{E_\xi}(X)+\gamma-\lambda(D^{E_\xi}(X)+\beta)\right\}\right.\right.$$

$$\left.\left.-\left\{N^{E_\xi}(X)+\gamma-\lambda_0(D^{E_\xi}(X)+\beta)\right\}\right\}\right|$$

$$\leq\left|\underset{X\in S}{Max}\left\{(\lambda_0-\lambda)(D^{E_\xi}(X)+\beta)\right\}\right|$$

$$\leq\left|\lambda_0-\lambda\right|\left|\underset{X\in S}{Max}(D^{E_\xi}(X)+\beta)\right|.$$

Let $0<\omega=\varepsilon/\eta$ then $\left|\lambda_0-\lambda\right|<\omega$, where $\eta=\left|\underset{X\in S}{Max}(D^{E_\xi}(X)+\beta)\right|$

$$\Rightarrow\left|\Lambda^{E_\xi}(X,\lambda)-\Lambda^{E_\xi}(X,\lambda_0)\right|\leq\left|\lambda_0-\lambda\right|\eta$$

$$\Rightarrow\left|\Lambda^{E_\xi}(X,\lambda)-\Lambda^{E_\xi}(X,\lambda_0)\right|\leq\omega\eta$$

Therefore $\left|\lambda-\lambda_0\right|<\omega\Rightarrow\left|\Lambda^{E_\xi}(X,\lambda)-\Lambda^{E_\xi}(X,\lambda_0)\right|<\varepsilon.$ □

**Property 4.2:** $\Lambda^{E_\xi}(X,\lambda)$ is strictly decreasing over the interval $\mathrm{I}\subset[0,\theta]$ and $X\subset\mathbb{R}^n$, where $\theta$ is an unknown parameter.

**Proof.** It is known that $\Lambda^{E_\xi}(X,\lambda)=\underset{X\in S}{Max}\left[(N^{E_\xi}(X)+\gamma)-\lambda(D^{E_\xi}(X)+\beta)\right]$. Let the variables $\lambda_1,\lambda_2\in\mathrm{I}\subset[0,\theta]\subset\mathbb{R}$ and also let $\lambda_1<\lambda_2$. Suppose $x_2\in S$ maximizes $\left[(N^{E_\xi}(X)+\gamma)-\lambda(D^{E_\xi}(X)+\beta)\right]$. Then,

$$\Lambda^{E_\xi}(X,\lambda_2) = \underset{X\in S}{Max}\Big[(N^{E_\xi}(X)+\gamma)-\lambda_2(D^{E_\xi}(X)+\beta)\Big]$$

$$= (N^{E_\xi}(x_2)+\gamma)-\lambda_2(D^{E_\xi}(x_2)+\beta)$$

$$< (N^{E_\xi}(x_2)+\gamma)-\lambda_1(D^{E_\xi}(x_2)+\beta)$$

$$\le \underset{X\in S}{Max}\Big[(N^{E_\xi}(X)+\gamma)-\lambda_1(D^{E_\xi}(X)+\beta)\Big] = \Lambda^{E_\xi}(X,\lambda_1)$$

It is clear that when $\lambda_1<\lambda_2$ then $\Lambda^{E_\xi}(X,\lambda_2)\le \Lambda^{E_\xi}(X,\lambda_1)$, therefore $\Lambda^{E_\xi}(X,\lambda)$ is strictly decreasing. □

It can be shown as in Charles and Dutta (2005b) that

(i) $X^*$ is optimal value of $R(X,\xi)$ if and only if $\Lambda^{E_\xi}(X^*,\theta^*)=0$, where $X^*$, $\theta^*$ are optimal values.

(ii) If $(X,\theta)\to(X^*,\theta^*)$, then $\Lambda^{E_\xi}(X,\theta)\to 0$, where $X^*$, $\theta^*$ are optimal values.

(iii) $\Lambda^{E_\xi}(X,\theta)>0$ for all $\theta<\theta^*$, where $\theta^*$ is optimal value.

## 4.5 HANDLING CHANCE CONSTRAINTS

Stochastic optimization algorithms normally incorporate probabilistic (random) elements in the objective function and/or constraints. When the constraints are easy to be handled, one can convert the chance constraints to their deterministic equivalents. But if the constraints fail to be regular or difficult to be handled, it is more convenient to deal with them by Monte-Carlo simulation techniques, which are applied to solve a good number of diverse problems in Science, Engineering and Business.

Monte-Carlo simulation technique is a problem solving technique used to approximate the probability of certain outcomes by running multiple trial runs, called simulation, using random variables. It mainly uses the concept of the Law of Large Numbers.

Now consider a chance constraint of the following form

$$\Pr\left[ g_i(X,\xi) \le 0 \right] \ge \alpha_i, \qquad 0 < \alpha_i < 1, \qquad i = 1, 2, \ldots, m$$

where $\xi = \{\xi_1, \xi_2, \ldots, \xi_t\}$ is a $t$- dimensional continuous random vector and each $\xi_i$ has a given distribution. For any given $X$, Rubinstein's (1981) Monte-Carlo simulation to estimate the above chance constraints (i.e., the value of the probabilistic constraints is the frequency with which the current solution $X$ satisfies the constraints) has been used.

In this algorithm, $N$ independent random vectors $\xi^{(p)} = \left( \xi_1^{(p)}, \xi_2^{(p)}, \ldots, \xi_t^{(p)} \right)$, $p = 1, 2, \ldots, N$, from their probability distribution have been generated and the generating methods have been discussed by numerous literature and summarized by Jana and Biswal (2004). Let $n_{\hat{g}_i}$ be the number of random vectors that satisfy the following chance constraints

$$g_i(X,\xi) \le 0, \qquad i = 1, 2, \ldots, m. \tag{4.7}$$

Then by the definition of probability, $\Pr\left[ g_i(X,\xi) \le 0 \right] = \frac{n_{\hat{g}_i}}{N}$. This means that chance constraints $\Pr\left[ g_i(X,\xi) \le 0 \right] \ge \alpha_i$ holds iff $\frac{n_{\hat{g}_i}}{N} \ge \alpha_i$. Certainly, this estimate is approximate and may change from one simulation to another.

## 4.6 GENETIC ALGORITHM FOR CCFP PROGRAMMING PROBLEMS

Genetic algorithms (GAs) are stochastic search methods based on the principles of natural genetic systems. They perform a search in providing an optimal solution for evaluation (fitness) function of an optimization problem. GAs deals simultaneously with multiple solutions and use only the fitness function values. While solving an optimization problem using GAs, each solution is coded as a string (called “chromosome”) of finite length. Each string or chromosome is considered as an individual. A collection of $K$ such individuals are called a population. GA starts with a randomly generated population of size $K$. In each generation, a new population of the same size is generated from the current population using three basic operations on the individuals of the population. The operators are (i) Selection (ii) Reproduction/Crossover and (iii) Mutation.

The new population obtained after selection, crossover and mutation is then used to generate another population. The best string in each generation is preserved either within the population or in a separate location outside the population. In this, the genetic algorithm would report the best value found, among all possible coded solutions obtained during the whole process.

GA has been well documented in the literature, such as in Koza (1992, 1994), Liu (2002). Recently Caramia and Onori (2008) experimented with a different crossover operator technique to solve the vehicle routing problem. Hill and Hiremath (2005) developed a new approach for generating an initial population which is stronger in terms of solution quality and solution diversity. The well-known applications of GA include Scheduling, Reliability design, Group technology, Pattern recognition, Expert system and several others in Mathematical programming. Stochastic vendor selection problem (Shiwei He et al 2009) and Static scheduling of multilevel assembly job shops problem (Omkumar et al 2009) are some of the recent ones in these applications.

### 4.6.1 Computational Framework Combining GA and Stochastic Simulation

In this section, an attempt has been made to design a GA in order to solve CCFP problems. The Representation structure, Initialization process, Evaluation function (Fitness function), Selection process, Crossover operation and Mutation operation have been discussed.

**Representation structure:** Let $V$ denote the set of chromosomes in the population

$v = \left(x_1, x_2, \ldots, x_n\right)$ be a chromosome in the population

$K$ be the population size

$M$ be the total number of generations that the population evolves and

$q$ denote a given generation.

**Initialization:** The individuals in the population are initialized using uniform random numbers that generate values within the pre-specified interval. The values of $x_j$ ($j$=1, 2,…, $n$) are chosen uniformly between 0 and the upper limits of the $j^{th}$ decision variable. The population size typically contains a few hundreds of possible solutions.

**Constraints checking by stochastic simulation:** Here the technique of stochastic simulation is employed, which was discussed in the previous section to check the constraints of the model.

**Fitness function:** The evolution of the possible solutions is guided by the fitness function, which is designed as a measure of the goodness of the solutions. There are different fitness evaluation schemes available, such as linear-fitness scaling and power-law scaling and ranking. Gen et al (1996) used the exponential fitness scaling scheme to avoid premature convergence and stalling of the solution. Since these are problem dependent, the fitness of each chromosome is calculated according to the objective function value.

**Selection:** The Selection procedure controls how individuals are chosen to mate and produce offspring in the following generation. The methods use various ways to bias the process so that fitter individuals have a greater probability of selection. Initially, Holland (1975) used the Roulette-Wheel selection strategy. The Tournament selection is given by Goldberg et al (1989). However both have some limitations. In this algorithm, a normalized geometric ranking method proposed by Joines and Houck (1994) and applied by Jana and Biswal (2004) has been used. In this method, the population has been stored according to the fitness values ranging from the best to the worst. Therefore, minimization and negativity of the objective function become eligible. The probability of the $i^{th}$ individual being selected in this method is defined by

$$\text{P(select the } i^{th} \text{ individual)} = a'(1-a)^{i-1} \tag{4.8}$$

where $a$ is the probability of selecting the best individual, $i$ is the rank of the individual,

$$a' = \frac{a}{\left[1-(1-a)^{p\_size}\right]} \tag{4.9}$$

and $p\_size$ is the size of the population.

**Crossover:** Crossover controls how two selected individuals are combined to produce new individuals /offspring. A parameter $p_c$ of a genetic system has been defined as the probability of crossover. Arithmetic crossover, which is defined as a convex combination of a pair of chromosomes, is adopted. i.e., for each pair of chromosomes in the current population, a random number $r$ is generated within (0,1); if $r < p_c$ , the given pair is selected for crossover.

**Mutation:** Mutation is an operator, which introduces small changes into the population to ensure variety and reduce the chance of local optima convergence. Similarly a parameter $p_m$ of a genetic system has been defined as the probability of mutation. For each chromosome, a random number $r$ is generated within (0,1); if $r < p_m$, the chromosome is selected for mutation in a free direction as given in Gen et al (1996).

Let $v = \left(x_1, x_2, \cdots, x_n\right)$ be the chromosome to be mutated and let a randomly generated direction of mutation be $d$. An offspring is made as follows:

$v' = v + \mu.d$ , where $\mu$ is a random real number. (4.10)

If the offspring is not feasible, then set $\mu$ by a random real number in $(0, \mu)$ until $v + \mu.d$ is feasible. The $q^{th}$ generation of the population is given by $V_q$ and the $k^{th}$ ($k$ =1, 2,.., $K$) chromosome in the population is given by $v_k$. The population size and the number of generations that vary linearly with the number of variables have been used.

**Stopping criteria:** The algorithm is terminated whenever the objective function values of the subsequent generations become approximately zero.

## 4.7 PROPOSED GENETIC ALGORITHM

In the SP literature (Kall and Wallace 1994), several researchers suggested various CCP models. The model coefficients of most of these models are assumed to follow independent normal distribution. This is because deriving the deterministic equivalent of the objective function and/or constraints of the model is a known factor. But in the chance constrained programming model, randomness is considered on both sides of chance constraints and hence the feasible set of the CCP problem is non-convex. It is well known that GAs proposed by Holland (1975) is capable of finding optimal solution to problems having non-convex feasible domain. Motivated by this, simulation based GA is proposed for solving CCFP problem.

**Proposed Algorithm**:

Step 1 : Initialize $\tau$ =1, a counter variable for the number of generations.

Step 2 : Randomly initialize $p\_size$ number of chromosomes according to the initialization process.

Step 3 : Calculate $\theta^{\tau} = \frac{N^E(X)+\gamma}{D^E(X)+\beta}$, where $X$ is any feasible chromosome generated in step 2.

Step 4 : Check the system constraints by the technique of stochastic simulation.

Step 5 : Update the chromosomes by crossover and mutation scheme as described above.

Step 6 : For all chromosomes, calculate the objective function value (fitness value) using $\Lambda^{E}(X,\theta^{\tau})$ = Max $[(N^{E}(X^{\tau})+\gamma)-\theta^{\tau}\ (D^{E}(X^{\tau})+\beta)]$.

Step 7 : Select chromosomes for the next generation according to the selection process.

Step 8 : Select the best chromosome $X^{\tau+1}$ in the current generation.

Step 9 : Calculate $\theta^{\tau+1}=(N^{E}(X^{\tau+1})+\gamma)/(D^{E}(X^{\tau+1})+\beta)$.

Step 10 : Repeat steps 4-9 by replacing $\theta^{\tau}$ by $\theta^{\tau+1}$ until all subsequent generations produce objective function values, which are approximately zero.

Step 11 : Report the best chromosome as the optimal solution.

The following Figure 3.1 shows the structure of the proposed GA.

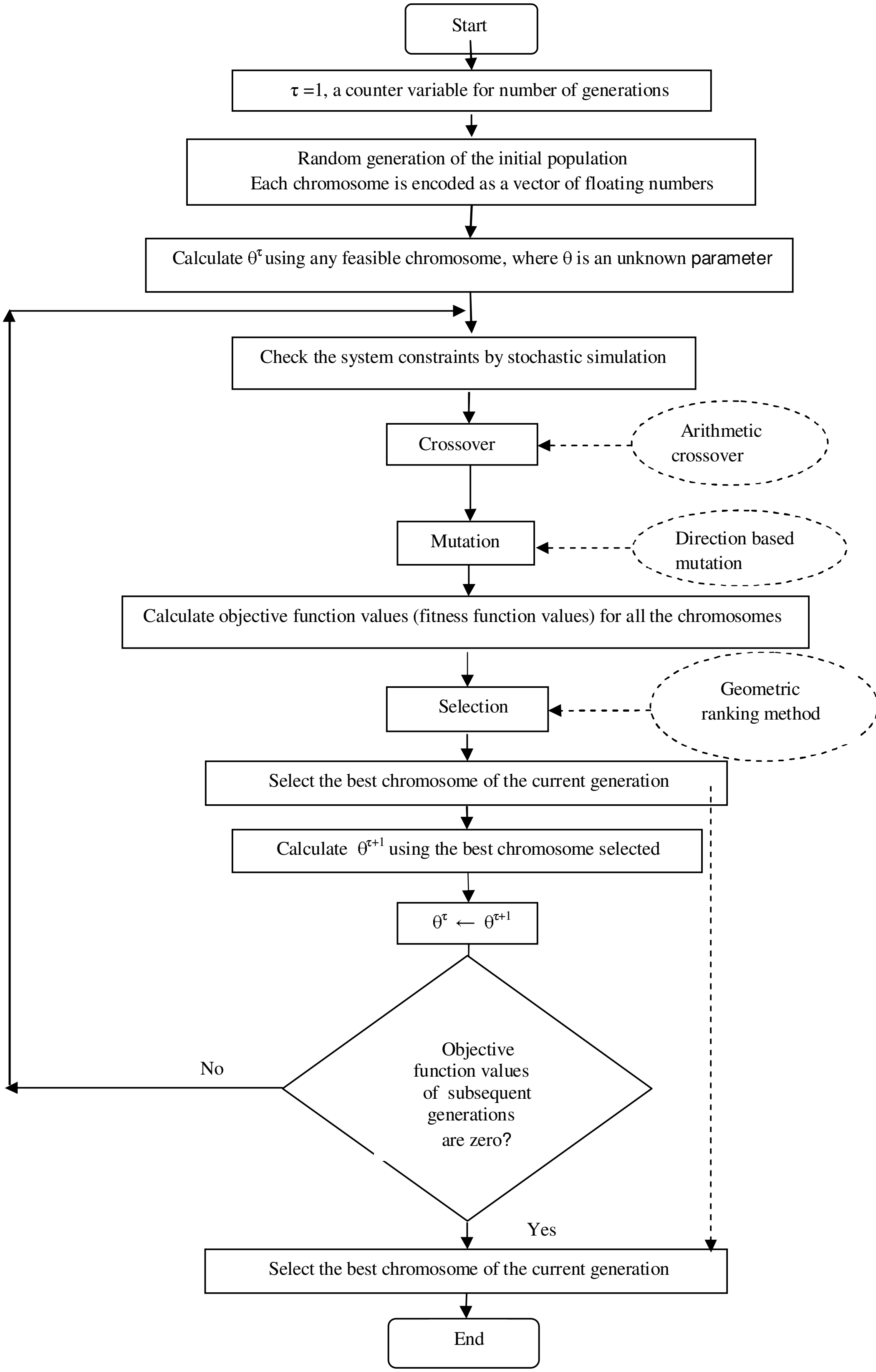


**Figure 4.1 Flowchart of the Proposed Algorithm**

## 4.8 NUMERICAL EXAMPLES

Most of the problem discussed in the literature of CCFP involves stochastic data following only Normal distributions. But in many managerial applications like transportation models, portfolio models, DEA models and so on, stochastic data can follow any continuous probability distribution. Hence in this section, a few test CCFP problems are formulated and solved. Here constraint coefficients and/or objective coefficients can follow continuous probability distributions such as Normal, Exponential and Weibull distributions.

The entire computation work has been done on a Pentium-III PC using the following parameters: the population size is 100, the probability of crossover is 0.8, the probability of mutation is 0.08 and the parameter $a$ in the normalized geometric ranking method is 0.005.

**Example 4.1**

Consider the following CCFP problem where there are two chance constraints with randomness only on the left hand side.

$$\text{Maximize} \quad (c_1 x_1 + c_2 x_2 + \gamma)/(d_1 x_1 + d_2 x_2 + \beta) \tag{4.11}$$

subject to

$$\Pr\left[ a_{11}x_1 + a_{12}x_2 \leq 10 \right] \geq \alpha_1$$

$$\Pr\left[ a_{21}x_1 + a_{22}x_2 \leq 8 \right] \geq \alpha_2$$

$$j = 1, 2$$

where $c_1 \sim N(2,1)$, $c_2 \sim N(3,1)$, $d_1 \sim N(3,1)$, $d_2 \sim N(5,2)$, $a_{11} \sim N(2,1)$, $a_{12} \sim N(3,2)$, $a_{21} \sim N(3,2)$, $a_{22} \sim N(1,0)$ and $\alpha_1 = \alpha_2 = 0.9, \gamma = 1,\ \beta = 2.$

The range used for decision variables is 0-2 and a run of stochastic simulation based genetic algorithm given in section 4.7 with 100 generations shows that the optimal solution of system (4.11) is $(x_1, x_2) = (1.6592, 0.0072)$ with the objective function value 0.6188. As shown in Figure 4.2, the fitness function (specially constructed objective function) values are tending towards zero approximately and hence the convergence of proposed GA.

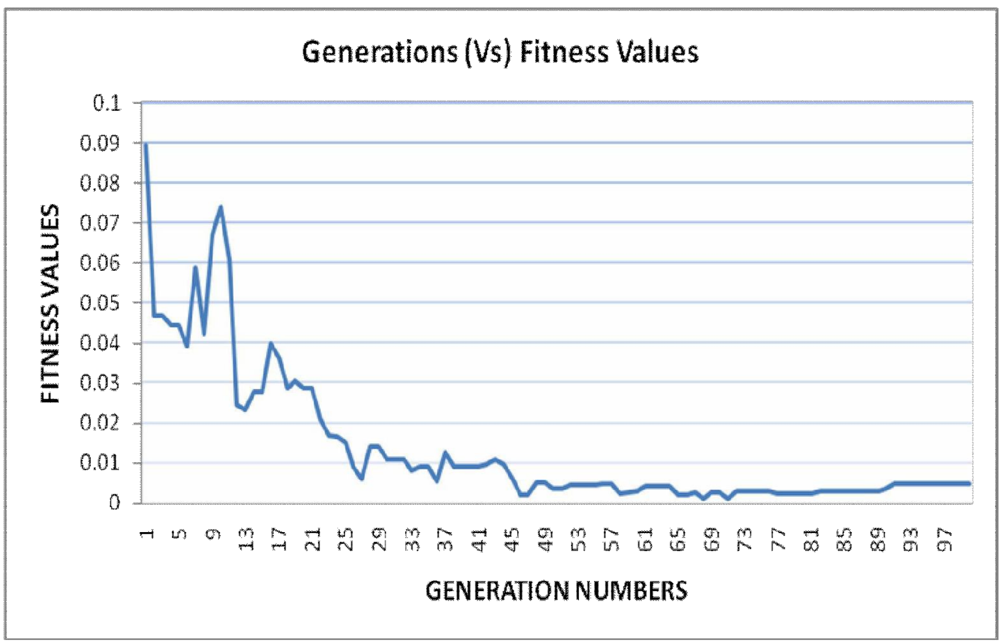


**Figure 4.2 Progress of GA Convergence for Example 4.1**

For comparing the GA results obtained above, the transformation technique of Charnes and Cooper (1962) and SLP for solving the problem defined in (4.11) has been adopted. The solution thus obtained is $(x_1, x_2) = (1.6631, 0.0000)$ with the objective function value of 0.6190.

**Example 4.2**

In the following CCFP, randomness is considered on only one side in the first and second constraints. In other words, constraints coefficients defined on the left-hand side of the first constraint are random and the resource element defined on the right hand side of the second constraint is also random. In the third constraint, interestingly randomness is considered on both sides of the constraint.

$$\text{Maximize} \quad (c_1 x_1 + c_2 x_2 + \gamma)/(d_1 x_1 + d_2 x_2 + \beta) \tag{4.12}$$

subject to

$$\Pr\left[a_{11}x_1 + a_{12}x_1 \leq 12\right] \geq \alpha_1$$

$$\Pr\left[2x_1 + x_2 \leq b_2\right] \geq \alpha_2$$

$$\Pr\left[a_{31}x_1 + a_{32}x_2 \leq b_3\right] \geq \alpha_3$$

$$j = 1, 2$$

where $c_1 \sim N(1, 0.5)$, $c_2 \sim N(1, 0.25)$, $d_1 \sim N(4, 3)$, $d_2 \sim N(1, 0)$, $a_{11} \sim N(4, 2)$, $a_{12} \sim N(2, 1)$, $a_{31} \sim N(1, 0.5)$, $a_{32} \sim N(1, 0.5)$, $b_2 \sim N(6, 2.5)$, $b_3 \sim N(4, 3)$ and $\alpha_1 = \alpha_2 = \alpha_3 = 0.9$, $\gamma = 0, \beta = 10$.

The range used for decision variables is 0-4 and a run of stochastic simulation based genetic algorithm given in section 4.7 with 100 generations shows that the optimal solution of system (4.12) is $(x_1, x_2) = (0.0376, 3.6413)$ with the objective function value 0.2667. As shown in Figure 4.3, the fitness function (specially constructed objective function) values are tending towards zero approximately and hence the convergence of proposed GA.

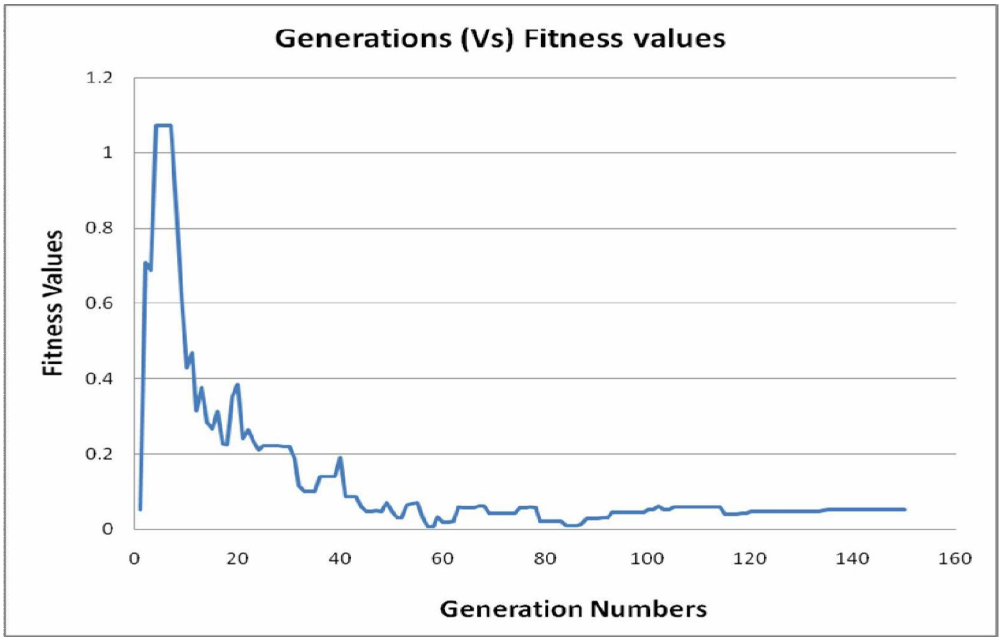


**Figure 4.3 Progress of GA Convergence for Example 4.2**

For comparing the GA results obtained above, the transformation technique of Charnes and Cooper (1962) and SLP for solving the problem defined in (4.12) has been adopted. The solution thus obtained is $(x_1, x_2) = (0.0000, 3.6585)$ with the objective function value of 0.2679.

**Example 4.3**

The following CCFP where chance constraint coefficients follow Exponential and Weibull distributions has been solved.

$$\text{Maximize } (c_1 x_1 + c_2 x_2 + \gamma)/(d_1 x_1 + d_2 x_2 + \beta) \tag{4.13}$$

subject to

$$\Pr\left[ a_{11}x_1 + a_{12}x_2 \leq 10 \right] \geq \alpha_1$$

$$\Pr\left[ a_{21}x_1 + a_{22}x_2 \leq 8 \right] \geq \alpha_2$$

$$j = 1, 2$$

where $c_1 \sim N(2,1)$, $c_2 \sim N(3,1)$, $d_1 \sim N(3,2)$, $d_2 \sim N(5,3)$. $a_{11}$, $a_{12}$ are Exponential random variables with mean values 2.5 and 1.3 respectively. $a_{21}, a_{22}$ are Weibull random variables with the parameters values: $a_{21} \sim W(1,2)$, $a_{22} \sim W(2,1.5)$ and $\alpha_1 = \alpha_2 = 0.9$, $\gamma = 1$, $\beta = 2$.

The range used for decisions variables is 0-2 and a run of stochastic simulation based genetic algorithm given in section 4.7 with 100 generations shows that the optimal solution of system (4.13) is $(x_1, x_2) = (1.7042, 0.0715)$ with the objective function value of 0.6467. As shown in Figure 4.4, the fitness function (specially constructed objective function) values are tending towards zero approximately and therefore global optimum is reached.

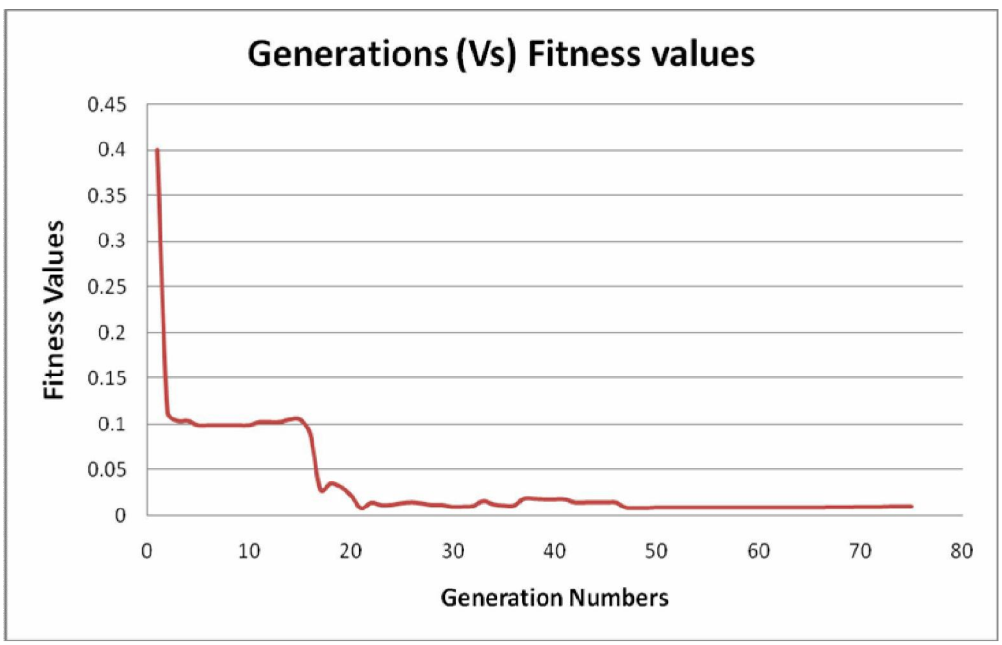


**Figure 4.4 Progress of GA Convergence for Example 4.3**

Getting closed form solution for this problem is highly difficult as some of the parameters follow Exponential and Weibull distributions and hence GA is used.

## 4.9 SUMMARY

In this chapter, CCFP model and its parametric form was introduced. The parametric form of CCFP problem was then solved by stochastic simulation based GA. Stochastic simulations have also been designed to check feasibility of stochastic constraints. The stochastic simulation based GA provides an effective means to solve complex CCFP problems. There are three advantages of using this GA: (i) To obtain global optima fairly well (ii) No need to convert the stochastic constraints into their deterministic equivalents where the translation is usually a hard task (ii) It can be used to solve a more general CCP problem. The effectiveness of stochastic simulation based GA has also been illustrated by a set of examples.