Logic Journal of IGPL Advance Access published February 8, 2011

Solving stochastic programming problems using modified differential evolution algorithms

RADHA THANGARAJ, *Faculty of Science, Technology and Communications, University of Luxembourg, rue Richard Coudenhove-Kalergi, Luxembourg.*
E-mail: t.radha@ieee.org

MILLIE PANT, *Department of Paper Technology, Indian Institute of Technology Roorkee, Saharanpur, Uttar Pradesh, India.*
E-mail: millifpt@iitr.ernet.in

PASCAL BOUVRY, *Faculty of Science, Technology and Communications, University of Luxembourg, rue Richard Coudenhove-Kalergi, Luxembourg.*
E-mail: pascal.bouvry@uni.lu

AJITH ABRAHAM, *Machine Intelligent Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, Seattle, Washington, USA.*
E-mail: ajith.abraham@ieee.org

Abstract

Stochastic (or probabilistic) programming (SP) is an optimization technique in which the constraints and/or the objective function of an optimization problem contain random variables. The mathematical models of these problems may follow any particular probability distribution for model coefficients. The objective here is to determine the proper values for model parameters influenced by random events. In this study, two modified differential evolution (DE) algorithms namely, LDE1 and LDE2 are used for solving SP problems. Two models of SP problems are considered; Stochastic Fractional Programming Problems and Multiobjective Stochastic Linear Programming Problems. The numerical results obtained by the LDE algorithms are compared with the results of basic DE, basic particle swarm optimization (PSO) and the available results from where it is observed that the LDE algorithms significantly improve the quality of solution of the considered problem in comparison with the quoted results in the literature.

Keywords: Differential evolution, stochastic programming, fractional programming, multiobjective optimization.

1 Introduction

Stochastic programming (SP) is a mathematical programming where stochastic element is present in the data. In contrast to deterministic mathematical programming where the data (coefficients) are known numbers, in SP these numbers follow a probability distribution. Thus, we can say that SP is a framework for modelling optimization problems that involve uncertainty. The goal here is to find some policy that is feasible for all (or almost all) the possible data instances and maximizes the expectation of some function of the decisions and the random variables. More generally, such

doi:10.1093/jigpal/jzr017

models are formulated, solved analytically or numerically and analysed in order to provide useful information to the decision-maker. SP has applications in a broad range of areas such as finance, transportation, energy optimization, etc. [4, 10, 13, 17].

SP provides a general framework to model path dependence of the stochastic process within an optimization model. Furthermore, it permits uncountably many states and actions, together with constraints, time lags, etc. One of the important distinctions that should be highlighted here is that unlike deterministic programming, SP separates the model formulation activity from the solution algorithm. An advantage of this separation is that it is not necessary for all SP models to obey the same mathematical assumptions. This leads to a rich class of models for which a variety of algorithms can be developed. On the other hand, SP formulations can lead to very large scale problems, and methods based on approximation and decomposition become paramount [16].

In the recent past, SP has been also applied to the problems having multiple, conflicting and non-commensurable objectives where generally there does not exist a single solution that can optimize all the objectives. Several methods for solving Multiobjective Stochastic Linear Programming (MOSLP) problems and their applications to various fields are available in literature [1–3, 7, 11, 12, 21]. Most of the probabilistic models assume normal distribution for model coefficients. Sahoo and Biswal [15] presented some deterministic equivalents for the probabilistic problems involving normal and log-normal random variables for joint constraints. Charles *et al.* [6] addressed different forms of distributions like Power Function distribution, Pareto distribution, Beta distribution of first kind, Weibull distribution and Burr type XII distribution.

In the present study, we have considered two types of SP problems. They are: (i) Stochastic Fractional Programming Problems (SFPPs) (ii) MOSLP problems. For SFPP, we have followed the models proposed by Charles and Dutta [8] and for MOSLP, the model proposed by Charles *et al.* [6] is followed. The aforesaid problems are solved using modified differential evolution (DE) algorithm called Laplace Differential Evolution (LDE), based on Laplace distribution; two versions of LDE algorithms (LDE1 and LDE2) are used in this study. The LDE algorithms are proposed by Thangaraj *et al.* [22] and analysed with standard benchmark problems and real-life problems. Encouraged by its performance, in the present study we have used the LDE algorithms for solving two SP models. The results obtained by LDE algorithms are compared with basic versions of DE and PSO and also with the results quoted in the literature [6, 8].

The rest of the article is organized as follows: Section 2 briefly describes the basic DE, LDE1 and LDE2 algorithms. The problem definitions are given in Section 3. In Section 4, the experimental settings and numerical results are discussed. Finally, the article concludes with Section 5.

2 DE algorithms

DE algorithm was developed by Storn and Price [19] in 1995. It is a novel evolutionary approach capable of handling non-differentiable, non-linear and multimodal objective functions. DE has been designed as a stochastic parallel direct search method, which utilizes the concepts borrowed from the broad class of Evolutionary Algorithms (EAs). It typically requires few, easily chosen control parameters. Experimental results have shown that the performance of DE is better than many other well-known EAs [18, 20]. While DE shares similarities with other EAs, it differs significantly in the sense that in DE, distance and direction information is used to guide the search process [9]. In this section, we give a brief introduction to the basic DE algorithm and its two modified versions LDE1 and LDE2.

2.1 Basic DE

A general DE variant may be denoted as *DE/X/Y/Z*, where X denotes the vector to be mutated, Y specifies the number of difference vectors used and Z specifies the crossover scheme which may be binomial (bin) or exponential (exp). Throughout the study, we shall consider the mutation strategy *DE/rand/1/bin* [19] which is perhaps the most frequently used version of DE. The operators used in this particular scheme are described as follows:

2.1.1 Mutation

For a D-dimensional search space, each target vector $X_{i,g}$, a mutant vector $V_{i,g}$ is generated by

$$V_{i,g+1} = X_{r_1,g} + F * (X_{r_2,g} - X_{r_3,g}) \tag{1}$$

where $r_1, r_2, r_3 \in \{1, 2, \ldots, \text{NP}\}$ are randomly chosen integers, different from each other and also different from the running index i.

$F(>0)$ is a scaling factor which controls the amplification of the DE $(X_{r_2,g} - X_{r_3,g})$.

2.1.2 Crossover

Once the mutation phase is over, crossover is performed between the target vector and the mutated vector to generate a trial point for the next generation. Crossover is introduced to increase the diversity of the population [20]. The mutated individual, $V_{i,g+1} = (v_{1,i,g+1}, \ldots, v_{D,i,g+1})$, and the current population member, $X_{i,g} = (x_{1,i,g}, \ldots, x_{D,i,g})$, are then subject to the crossover operation, that finally generates a population of candidate solutions or 'trial' vectors, $U_{i,g+1} = (u_{1,i,g+1}, \ldots, u_{D,i,g+1})$, as follows:

$$u_{j,i,g+1} = \begin{cases} v_{j,i,g+1} & \text{if} \quad \text{rand}_j \le C_r \vee j = j_{\text{rand}} \\ x_{j,i,g+1} & \text{otherwise} \end{cases} \tag{2}$$

where $j = 1, 2, \ldots, D$; $\text{rand}_j \in [0, 1]$;

C_r is the called the crossover constant and it takes values in the range [0, 1] $j_{\text{rand}} \in (1, 2, \ldots, D)$ is a randomly chosen index.

2.1.3 Selection

The final phase of DE algorithm is selection. Here, the population for the next generation is selected from the individual in current population and its corresponding trial vector according to the following rule:

$$X_{i,g+1} = \begin{cases} U_{i,g+1} & \text{if} \quad f(U_{i,g+1}) \le f(X_{i,g}) \\ X_{i,g} & \text{otherwise} \end{cases} \tag{3}$$

Thus, each individual of the advance (trial) population is compared with its counterpart in the current population. The one with the lower objective function value will survive the tournament selection to form the population for the next generation. As a result, all the individuals of the next generation are as good as or better than their counterparts in the current generation.

2.2 LDE

The LDE algorithms proposed by Thangaraj *et al.* [22] are simple and modified version of the basic DE algorithm. First, the LDE schemes make use of the absolute weighted difference between the two vector points which is in contrast to the basic DE, where the usual vector difference is considered. Secondly, in LDE schemes the amplification factor, F (of the usual DE), is replaced by L, a random variable following Laplace distribution.

The Probability Density Function (pdf) of Laplace distribution is similar to that of normal distribution; however, the normal distribution is expressed in terms of squared difference from the mean while Laplace density is expressed in terms of absolute difference from the mean. The density function of Laplace distribution is given as:

$$f(x/\theta)=\frac{1}{2\mu}\exp\left(\frac{-|x-\theta|}{\mu}\right),\ -\infty\leq x\leq\infty \tag{4}$$

Its distribution function is given by:

$$==\frac{1}{2\mu}\begin{cases}\exp\left(-\frac{x-\theta}{\mu}\right) & \text{if} \quad x\leq\theta \\ 1-\exp\left(-\frac{\theta-x}{\mu}\right) & \text{if} \quad x>0\end{cases} \tag{5}$$

$\mu>0$ is the scale parameter.

The mutation schemes of LDE1 and LDE2 algorithms are defined as follows:

(i) LDE1 scheme

$$v_{i,g+1}=x_{\text{best},g}+\text{L}*|x_{r_1,g}-x_{r_2,g}| \tag{6}$$

In LDE1 scheme, the base vector is the one having the best fitness function value; whereas, the other two individuals are randomly selected.

(ii) LDE2 scheme
If $(U(0,1)<0.5)$ then $v_{i,g+1}=x_{\text{best},g}+\text{L}*|x_{r_1,g}-x_{r_2,g}|$
Else $v_{i,g+1}=x_{r_1,g}+F*(x_{r_2,g}-x_{r_3,g})$

In LDE2 scheme, mutant vector using Equation (6) and the basic mutant vector equation are applied probabilistically using a predefined value. A random variable following normal distribution $U(0,\,1)$ is generated. If it is less than 0.5, then LDE1 scheme is applied otherwise Equation (1) is applied.

From the above said schemes, it can be seen that the newly generated mutant vector will lie in the vicinity of the base vector. However, its nearness or distance from base vector will be controlled by L.

For smaller values of L, the mutant vector is likely to be produced near the initially chosen vector, whereas for larger values of L, the mutant vector is more likely to be produced at a distance from the chosen vector. This behaviour makes the algorithm self-adaptive in nature.

Both the modified versions, LDE1 and LDE2, have given good performances for solving benchmark as well as real-life problems [22].

3 Problem definition

This section is divided into two subsections; in Section 3.1, the problem formulation of linear SFPPs is given and the general model of the multiobjective SP problems with two test examples is given in Section 3.2.

3.1 Linear Stochastic Fractional Programming Model

A linear stochastic fractional programming (LSFP) problem involves optimizing the ratio of two linear functions subject to some constraints in which at least one of the problem data is random in nature with non-negative constraints on the variables. Additionally, some of the constraints may be deterministic [5]. The LSFP framework attempts to model uncertainty in the data by assuming that the input or a part thereof is specified by a probability distribution, rather than being deterministic. The problem of optimizing sum of more than one ratios of function is called stochastic sum-of-probabilistic fractional programming (SSFP) problem when the data under study are random in nature. The following section gives the general model of the SSFP problems.

3.1.1 General Model of SSFP problem

The mathematical model of a stochastic SSFP problem can be expressed as follows [8]:

$$\operatorname*{Max}_{x\in S} R(X)=\sum_{y=1}^{k} R_y(X),$$

where $R_y(X)=\frac{N_y(X)+\alpha_y}{D_y(X)+\beta_y}$, $y=1,2,\ldots,k$

Subject to:

$$P\left(\sum_{j=1}^{n} t_{ij}x_j \le b_i^{(1)}\right) \ge 1-p_i^{(1)}, \quad i=12,\ldots,m; \tag{7}$$

$$\sum_{j=1}^{n} t_{ij}x_j \le b_i^{(2)}, \quad i=m+1,\ldots,h \tag{8}$$

where $0\le X_{nx1}=\left\|x_j\right\| \subset R^n$ is a feasible set and $R:R^n\to R^k$,

$$T_{mxn}=\left\|t_{ij}^{(1)}\right\|,$$

$$b_{mx1}^{(1)}=\left\|b_i^{(1)}\right\|, \quad i=1,2,\ldots,m, \quad j=1,2,\ldots,n;$$

$$b_{h-(m+1)x1}^{(2)}=\left\|b_i^{(2)}\right\|, \quad i=m+1,\ldots,h; \quad \alpha_y,\ \beta_y \text{ are scalars.}$$

$$N_y(X)=\sum_{j=1}^{n} c_{yj}x_j \text{ and } D_y(X)=\sum_{j=1}^{n} d_{yj}x_j.$$

In this model, out of $N_y(X)$, $D_y(X)$, T and $b^{(1)}$ at least one may be a random variable.
$S=\{X|\text{Equation (7)}-(8),\ X\ge 0,\ X\subset R^n\}$ is non-empty, convex and compact set in R^n.

3.1.2 Test Example 1 (SSFP1)

$$\text{Max } R(X)=\sum_{y=1}^{2}\frac{c_{y1}x_1+c_{y2}x_2+\alpha_y}{d_{y1}x_1+d_{y2}x_2+\beta_y}$$

Subject to: $a_{11}x_1+a_{12}x_2 \le 1$, $a_{21}x_1+a_{22}x_2 \le b_2$, $16x_1+x_2 \le 4$, $x_1, x_2 \ge 0$

The deterministic model of the above problem may be given as:

$$\text{Max } \lambda_1+\lambda_2$$

Subject to: $(\lambda_1+2\lambda_2-5)x_1+(\lambda_1+3\lambda_2-4)x_2+2\lambda_1+4\lambda_2+1.28\sqrt{x_1^2+x_2^2} \le 3$, $(2x_1+x_2)+1.645\sqrt{x_1^2+x_2^2} \le 1$, $(3x_1+4x_2)+0.84\sqrt{2x_1^2+3x_2^2+2} \le 3$, $16x_1+x_2 \le 4$, $x_1, x_2, \lambda_1, \lambda_2 \ge 0$.

3.1.3 Test example 2 (SSFP2)

$$\text{Max } R(X)=\sum_{y=1}^{3}\frac{c_{y1}x_1+c_{y2}x_2+\alpha_y}{d_{y1}x_1+d_{y2}x_2+\beta_y}$$

Subject to: $a_{11}x_1+a_{12}x_2++a_{13}x_3 \le b_1$, $a_{31}x_1+a_{32}x_2+a_{33}x_3 \le 20$, $x_1+x_2+x_3 \le b_3$,

$$5x_1+3x_2+4x_3 \le 15, \quad x_1, x_2, x_3 \ge 0$$

The deterministic model of the above problem is:

$$\text{Max } \lambda_1+\lambda_2+\lambda_3$$

Subject to:

$$(\lambda_1+2\lambda_2+4\lambda_3-17)x_1+(\lambda_1+\lambda_2+3\lambda_3-19)x_2+(\lambda_1+4\lambda_2+7\lambda_3-23)x_3+2\lambda_1+10\lambda_2+5\lambda_3$$
$$+1.645\sqrt{(\lambda_2^2+0.5\lambda_3^2)x_1^2+(0.5\lambda_2^2+2\lambda_3^2)x_2^2+(2\lambda_2^2+3\lambda_3^2)x_3^2} \le 12$$
$$4x_1+2x_2+4x_3+1.645\sqrt{0.5x_1^2+0.25x_2^2+0.5x_3^2+0.25} \le 12,$$
$$6x_1+4x_2+6x_3+1.28\sqrt{x_1^2+0.5x_2^2+0.75x_3^2} \le 20, \quad x_1+x_2+x_3 \le 3.16,$$
$$5x_1+3x_2+4x_3 \le 15, \quad x_1, x_2, x_3, \lambda_1, \lambda_2, \lambda_3 \ge 0.$$

3.1.4 Test example 3 (SSFP3)

$$\text{Max } R(X)=\sum_{y=1}^{2}\frac{c_{y1}x_1+c_{y2}x_2+\alpha_y}{d_{y1}x_1+d_{y2}x_2+\beta_y}$$

Subject to: $a_{11}x_1+a_{12}x_2++a_{13}x_3 \le 27$, $5x_1+3x_2+x_3 \le 12$, $x_1, x_2, x_3 \ge 0$

The deterministic model of the above problem is:

$$\text{Max } \lambda_1+\lambda_2$$

Subject to:

$$(20-2\lambda_1-4\lambda_2)x_1+(16-3\lambda_1-2\lambda_2)x_2+(12-5\lambda_1-2\lambda_2)x_3-10\lambda_1-12\lambda_2$$
$$-1.28\sqrt{(\lambda_1^2+\lambda_2^2+10)x_1^2+(2\lambda_1^2+\lambda_2^2+4)x_2^2+(3\lambda_1^2+2\lambda_2^2+5)x_3^2} \ge 3$$
$$3x_1+4x_2+8x_3+1.645\sqrt{2x_1^2+x_2^2+x_3^2} \le 27, \quad 5x_1+3x_2+x_3 \le 12, \quad x_1, x_2, x_3, \lambda_1, \lambda_2 \ge 0.$$

For more details on the above examples, please refer [8].

3.2 MOSLP

The mathematical model of the MOSLP problem used in the present study is given in the following subsection.

3.2.1 General Model

The general mathematical model of a constrained MOSLP may be given as [6]:

$$\text{Maximize } z_k = \sum_{j=1}^{n} c_j^k x_j, \quad k = 1, 2, \ldots, K$$

Subject to

$$P\left(\sum_{j=1}^{n} a_{1j}x_j \le b_1, \sum_{j=1}^{n} a_{2j}x_j \le b_2, \ldots, \sum_{j=1}^{n} a_{mj}x_j \le b_m\right) \ge p, x_j \ge 0, j = 1, 2, \ldots, n$$

where $0 < p < 1$ is usually close to 1. It has been assumed that the parameters a_{ij} and c_j are deterministic constants and b_i are random variables. For more details, the interested reader may please refer to [6].

3.2.2 Test example 1(MOSLP1)

$$\text{Maximize } z_1 = 5x_1 + 6x_2 + 3x_3, \quad \text{Maximize } z_2 = 6x_1 + 3x_2 + 5x_3,$$
$$\text{Maximize } z_3 = 2x_1 + 5x_2 + 8x_3$$

Subject to

$$P(3x_1 + 2x_2 + 2x_3 \le b_1) \ge 0.90, \quad P(2x_1 + 8x_2 + 5x_3 \le b_2) \ge 0.98, \quad P(5x_1 + 3x_2 + 2x_3 \le b_3) \ge 0.95,$$
$$P(0.5x_1 + 0.5x_2 + 0.25x_3 \le b_4) \ge 0.90, \quad P(8x_1 + 3x_2 + 4x_3 \le b_5) \ge 0.99, \quad x_1, x_2, x_3 \ge 0$$

Here, b_1 follow Power Function distribution, b_2 follow Pareto distribution, b_3 follow Beta distribution, b_4 follow Weibull distribution; b_5 follow Burr type XII distribution. The problem is converted to deterministic model as follows:

$$\text{Maximize } z = \lambda_1(5x_1 + 6x_2 + 3x_3) + \lambda_2(6x_1 + 3x_2 + 5x_3) + \lambda_3(2x_1 + 5x_2 + 8x_3)$$

Subject to

$$3x_1 + 2x_2 + 2x_3 \le 6.3096, \quad 2x_1 + 8x_2 + 5x_3 \le 8.0812, \quad 5x_1 + 3x_2 + 2x_3 \le 4.7115,$$
$$0.5x_1 + 0.5x_2 + 0.25x_3 \le 0.9379, \quad 8x_1 + 3x_2 + 4x_3 \le 10.0321, \quad \lambda_1 + \lambda_2 + \lambda_3 = 1$$
$$x_1, x_2, x_3, \lambda_1, \lambda_2, \lambda_3 \ge 0$$

3.2.3 Test example 2 (MOSLP2)

$$\text{Maximize } z_1 = 3x_1 + 8x_2 + 5x_3, \quad \text{Maximize } z_2 = 7x_1 + 4x_2 + 3x_3$$
$$\text{Maximize } z_3 = 6x_1 + 7x_2 + 10.5x_3$$

Subject to

$$P(5x_1 + 4x_2 + 2x_3 \le b_1) \ge 0.95, \quad P(7x_1 + 3x_2 + x_3 \le b_2) \ge 0.95, P(2x_1 + 7x_2 + 3x_3 \le b_3) \ge 0.95,$$
$$P(2x_1 + 3x_2 + 2.5x_3 \le b_4) \ge 0.95, \quad P(5x_1 + 2x_2 + 1.5x_3 \le b_5) \ge 0.95,$$
$$x_1, x_2, x_3 \ge 0$$

Here b_1 follow Power Function distribution; b_2 follow Pareto distribution; b_3 follow Beta distribution of first kind; b_4 follow Weibull distribution and b_5 follow Burr type XII distribution. The deterministic model of the problem is given as:

$$\text{Maximize } z = \lambda_1(3x_1 + 8x_2 + 5x_3) + \lambda_2(7x_1 + 4x_2 + 3x_3) + \lambda_3(6x_1 + 7x_2 + 10.5x_3)$$

Subject to

$$\left[\frac{y_1^2}{9}\right]\left[\frac{y_2^2 - 100}{y_2^2}\right]\left[\frac{y_3 - 5}{10}\right]\left[\frac{e^{2y_4} - 1}{e^{2y_4}}\right]\left[\frac{3y_5^2}{1 + 3y_5^2}\right] \ge 0.95$$
$$5x_1 + 4x_2 + 2x_3 = y_1, \quad 7x_1 + 3x_2 + x_3 = y_2, \quad 2x_1 + 7x_2 + 3x_3 = y_3, \quad 2x_1 + 3x_2 + 2.5x_3 = y_4,$$
$$5x_1 + 2x_2 + 1.5x_3 = y_5, \quad \lambda_1 + \lambda_2 + \lambda_3 = 1 x_1, x_2, x_3, y_1, y_2, y_3, y_4, y_5, \lambda_1, \lambda_2, \lambda_3 \ge 0$$

4 Experimental settings and numerical results

4.1 Parameter settings

The three main control parameters of DE population size, crossover rate Cr and scaling factor F are fixed as 50, 0.5 and 0.5, respectively. For LDE schemes, the scaling factor is a random variable, L, following Laplace distribution. For each algorithm, the stopping criterion is to terminate the search process when one of the following conditions is satisfied:

(i) the maximum number of generations is reached (assumed 1000 generations),
(ii) $|f_{\max} - f_{\min}| < 10-^4$ where f is the value of objective function.

Constraints are handled according to the approach based on repair methods suggested in [14]. A total of 50 runs for each experimental setting were conducted and the best solution throughout the run was recorded as global optimum. Results obtained by the LDE versions are compared with basic DE, basic PSO and also previously quoted results [6, 8].

To solve the given problems by PSO, we have adopted the following settings for its control parameters: inertia weight w: linearly decreasing from 0.9 to 0.4; acceleration coefficients $c_1 = c_2 = 2.0$.

4.2 Numerical results

LDE schemes are compared with basic DE and basic PSO through various performance metrics like average fitness function value and standard deviation (SD). To compare the convergence speed of algorithms, we considered the average number of function evaluations (NFEs).

TABLE 1. Results of stochastic sum-of-fractional programming problems

	DE	LDE1	LDE2	PSO	Results in [8]
SSFP1					
F(x)	1.83245	1.83246	1.83246	1.83218	1.7533
R(X)	1.45685	1.45685	1.45687	1.45676	1.40035
x_1	0.202128	0.20199	0.202329	0.20254	0.0602
x_2	0.165738	0.165968	0.165426	0.164923	0.3292
λ_1	1.83245	1.83246	1.83246	1.83218	1.7533
λ_2	0	0	0	0	0
SD	1.2639e-5	1.2143e-5	1.2147e-5	0.120788	NA
Average NFE	15925	11790	5455	37269	NA
SSFP2					
F(x)	15.2256	15.0329	15.2256	15.2231	15.1931
R(X)	5.29363	5.5283	5.30394	5.32662	5.29317
x_1	0	0.0101852	0	0	0
x_2	1.42539	1.76232	1.43882	1.46877	1.4248
x_3	1.68131	1.3875	1.67478	1.65862	1.6816
λ_1	15.2256	15.0329	15.2256	15.2231	15.1931
λ_2	0	7.278e-8	7.268e-8	0	0
λ_3	6.039e-7	5.641e-12	0	0	0
SD	1.4346e-5	1.2029e-5	1.1060e-5	1.06941	NA
Average NFE	38855	4540	10460	42352	NA
SSFP3					
F(x)	2.32854	2.23663	2.33083	2.23657	3.6584
R(X)	3.41456	3.53474	3.59134	3.53478	3.5348
x_1	1.88002	2.39981	1.781	2.4	2.4
x_2	0.866625	0.00031	1.03165	0	0
x_3	0	0	0	0	0
λ_1	2.32854	2.23663	2.33083	2.23657	3.6584
λ_2	2.095e-7	1.923e-6	0	0	0
SD	0.05265	0.03543	0.03305	0.335365	NA
Average NFE	5725	5350	4850	44656	NA

4.2.1 Results analysis of SSFP problems

Performance comparison of LDE algorithms with basic DE and basic PSO are given in Table 1 in terms of objective function value of deterministic model (F(X)), objective function value of stochastic model (R(X)), decision variable values, standard deviation and average NFEs. From the numerical results of Table 1, we can see that LDE algorithms are superior to the other algorithms. For the first test problem SSFP1, the improvement percentage of LDE1 in terms of NFE in comparison to DE and PSO are 25.9 and 68.36%, respectively, whereas the improvement % of LDE2 algorithm in comparison to DE and PSO are 65 and 85.36%, respectively. Similarly for the remaining two test problems also the LDE algorithms gave a noticeable percentage of improvement in terms of objective function value and NFE in comparison to basic DE, PSO and the quoted results in the literature. Figure 1 shows the performance of Laplace DE algorithm for all the SSFPs and MOSLPs problems in terms of objective function values.

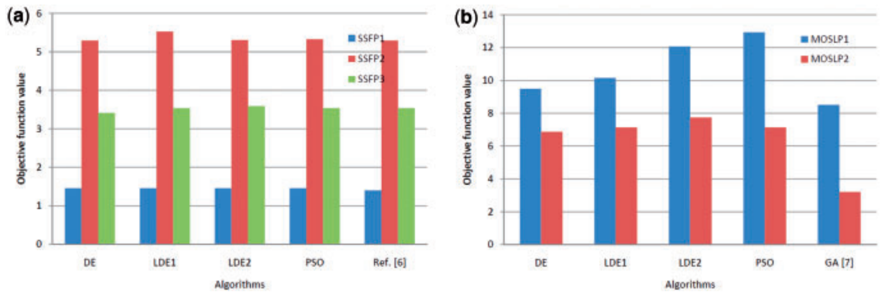

FIG. 1. Performance of LDE, DE and PSO algorithms in terms of objective function value (**a**) for SSFPs (**b**) for MOSLPs.

4.2.2 Results analysis of MOSLP problems

We have considered four test cases in each of the test problems MOSLP1 and MOSLP2. Since, $\lambda_1+\lambda_2+\lambda_3=1$, one of λ_i, $i=1,2,3$ could be eliminated to reduce the number of dependent variables from the expression of objective function. So, we assigned equal weights to two terms at a time in the objective expression. The resultant test cases are as follows:

(i) $\lambda_1=W, \lambda_2=\lambda_3=\frac{1-W}{2}, 0\leq W\leq 1$
(ii) $\lambda_2=W, \lambda_1=\lambda_3=\frac{1-W}{2}, 0\leq W\leq 1$
(iii) $\lambda_3=W, \lambda_1=\lambda_2=\frac{1-W}{2}, 0\leq W\leq 1$
(iv) λ_1, λ_2 and λ_3 are dependent variables.

The numerical results of MOSLP1 and MOSLP2 are recorded in Tables 2 and 3, respectively. The best solution obtained by LDE, DE and PSO algorithms for MOSLP1 in terms of optimal decision variable values, objective function value, standard deviation and average NFEs is given in Table 2. For the first three test cases, the LDE algorithms are superior to others in terms of objective function value; if we compare the LDE algorithms with each other then from the results it can be clear that LDE2 algorithm is better than LDE1 algorithm. For the test case (iv), the performance of PSO is better than all the other algorithms in terms of objective function value. If we compare the algorithms with respect to NFE, then from Table 2 it can be seen that LDE1 gave good result. There is an improvement of 52% in objective function value when the problem is solved by LDE2 in comparison with the quoted result [6], where the problem is solved by Genetic Algorithm (GA). Similarly the improvement % of LDE1 algorithm is 19.3%. The results of test problem MOSLP2 are given in Table 3. From this table also we can see that LDE2 algorithm is superior with others in all the test cases. The improvement of LDE2 algorithm in comparison with the results in the literature is 141%. Figure 1 shows the performance of LDE, DE and PSO algorithms in terms of objective function value.

5 Conclusions

SP is an optimization technique in which the constraints and/or the objective function of an optimization problem contains certain random variables following different probability distributions. In the present study, two models of SP problems were considered; (i) SFPPs and (ii) MOSLP Problems. The test problems were solved using the modified DE algorithm called Laplace DE algorithm (LDE).

TABLE 2. Results of MOSLP1

	DE	LDE1	LDE2	PSO	GA [6]
$\lambda_1 = W$, $\lambda_1 = \lambda_2 = (1-W)/2$, $0 \le W \le 1$					
z	10.9905	10.996	10.997	10.9887	NA
z_1	6.17418	6.18472	6.18566	6.18385	
z_2	9.47566	9.48721	9.48832	9.48557	
z_3	12.5077	12.5065	12.5068	12.5033	
x_1	0.349128	0.35171	0.351905	0.351792	
x_2	0	0	0	0	
x_3	1.47618	1.47539	1.47538	1.47496	
SD	0.009314	0.001586	0.001135	0.224631	
Average NFE	50050	50050	50050	50050	
$\lambda_2 = W$, $\lambda_1 = \lambda_3 = (1-W)/2$, $0 \le W \le 1$					
z	9.48974	9.48975	9.48975	9.48963	NA
z_1	6.18684	6.18686	6.18687	6.18679	
z_2	9.48975	9.48975	9.48977	9.48965	
z_3	12.5073	12.50726	12.5073	12.5072	
x_1	0.35214	0.35215	0.352142	0.352143	
x_2	0	0	0	0	
x_3	1.47538	1.47537	1.47538	1.47536	
SD	0.000292	1.60188	0.723669	0.280235	
Average NFE	50050	50050	49578	50050	
$\lambda_3 = W$, $\lambda_1 = \lambda_2 = (1-W)/2$, $0 \le W \le 1$					
z	12.9277	12.9288	12.9292	12.9287	NA
z_1	4.84834	4.84836	4.84851	4.84834	
z_2	8.08056	8.08059	8.08085	8.08057	
z_3	12.9289	12.929	12.9295	12.9289	
x_1	0	0	0	0	
x_2	0	0	0	0	
x_3	1.61611	1.61612	1.61617	1.61611	
SD	0.009056	0.02954	0.01799	0.003374	
Average NFE	50050	50050	50050	50050	
Problem described as in [6]					
z	9.48978	10.1553	12.0647	12.9299	8.5089
z_1	6.18688	6.22744	6.12031	4.84872	6.4834
z_2	9.48978	9.34841	9.39769	8.0812	8.3125
z_3	12.5073	12.2764	12.4214	12.9299	10.5140
x_1	0.352147	0.35354	0.344071	0	0.3727
x_2	2.124e-7	0.0293907	0	0	0.2319
x_3	1.47538	1.4278	1.46665	1.61624	1.0761
SD	2.06789	1.62183	1.95275	3.91715	NA
Average NFE	14691	6932	7250	20573	NA

TABLE 3. Results of MOSLP2

	DE	LDE1	LDE2	PSO	GA [6]
$\lambda_1 = W$, $\lambda_1 = \lambda_2 = (1-W)/2$, $0 \leq W \leq 1$					
z	5.5452	6.3844	6.86328	7.29299	NA
z_1	4.60113	4.48698	4.56284	4.9091	
z_2	3.61702	4.0707	4.28202	4.36365	
z_3	9.25069	8.69822	9.4451	10.2273	
x_1	0.170342	0.275175	0.297729	0.272728	
x_2	0.0367932	0.0654974	0.00485206	0	
x_3	0.759151	0.627495	0.726168	0.818182	
y_1	2.5158	2.89285	2.96039	3.0	
y_2	2.06291	2.7502	2.82483	2.72728	
y_3	2.862	2.89131	2.80793	3.0	
y_4	2.36484	2.31558	2.42544	2.59091	
y_5	2.06754	2.44811	2.5876	2.59091	
SD	1.61902	1.44067	1.4189	2.3473	
Average NFE	50050	50050	50050	50050	
$\lambda_2 = W$, $\lambda_1 = \lambda_3 = (1-W)/2$, $0 \leq W \leq 1$					
z	5.3215	7.01255	7.74695	7.72732	NA
z_1	4.60113	4.71818	4.99999	4.99081	
z_2	3.61702	3.42226	3.0	2.99315	
z_3	9.25069	9.30713	10.5	10.4644	
x_1	0.170342	0.12258	0	0	
x_2	0.0367932	0.0575791	0	0.00166162	
x_3	0.759151	0.777962	0.999999	0.995503	
y_1	2.5158	2.39914	2.0	1.99765	
y_2	2.06291	1.80875	1.00001	1.00048	
y_3	2.862	2.98209	3.0	2.99814	
y_4	2.36484	2.36281	2.49999	2.49374	
y_5	2.06754	1.895	1.50001	1.49658	
SD	1.62021	1.60347	2.42642	1.60188	
Average NFE	50050	50050	50050	50050	
$\lambda_3 = W$, $\lambda_1 = \lambda_2 = (1-W)/2$, $0 \leq W \leq 1$					
z	6.60213	9.3271	10.4638	7.89437	NA
z_1	4.60113	4.45956	4.99082	5.06437	
z_2	3.61702	3.33100	2.99316	3.03862	
z_3	9.25069	9.32727	10.4644	10.6352	
x_1	0.170342	0.126015	0	0	
x_2	0.0367932	0	0.00166304	0	
x_3	0.759151	0.816303	0.995504	1.01287	
y_1	2.5158	2.26268	1.99765	2.02575	
y_2	2.06291	1.69841	1.00049	1.01288	
y_3	2.862	2.70093	2.99815	3.0	

(*Continued*)

TABLE 3 Continued

	DE	LDE1	LDE2	PSO	GA [6]
y_4	2.36484	2.29278	2.49374	3.0	
y_5	2.06754	1.85453	1.49659	1.51931	
SD	2.03924	2.38526	2.4733	3.12773	
Average NFE	50050	50050	50050	50050	
Problem described as in [6]					
z	6.87235	7.13425	7.73912	7.13425	3.2081
z_1	3.32379	4.18588	4.46621	4.18588	3.8139
z_2	1.99418	2.46762	2.57927	2.46762	3.0717
z_3	6.9787	8.19198	8.12872	8.19198	5.1968
x_1	2.65138e-006	0.000944931	0.000308158	0.000944931	0.1939
x_2	0.000127494	0.061029	0.127573	0.061029	0.2810
x_3	0.664552	0.738963	0.688939	0.738963	0.1968
y_1	1.32963	1.72678	1.88971	1.72678	2.4872
y_2	0.664947	0.928675	1.07383	0.928675	2.3971
y_3	1.99454	2.64598	2.96046	2.64598	2.9454
y_4	1.66177	2.03239	2.10569	2.03239	1.7229
y_5	0.9971	1.0	1.0	1.0	1.8267
SD	2.20171	1.77604	1.41855	0.490581	NA
Average NFE	30794	50050	50050	50050	NA

For the MOSLP test problems, four test cases were considered with respect to the weighing factors and the results were produced (for both SSFP and MOSLP) in terms of objective function value, decision variable values, standard deviation and average number function evaluations. The results of LDE algorithms were compared with the results of basic DE, PSO and the quoted results from which it is observed that the LDE algorithms significantly improve the quality of solution of the considered SP problems. This shows that the LDE algorithms provide an attractive alternative for solving SP problems.

Acknowledgement

This work was carried out during the tenure of an ERCIM 'Alain Bensoussan' Fellowship Programme.

References

[1] F. B. Abdelaziz, B. Aouni, and F. EI. Rimeh. Multi-objective programming for portfolio selection. *European Journal Operational Research*, **177**, 1811–1823, 2007.

[2] N. Baba and A. Morimoto. Stochastic approximations methods for solving the stochastic multi-objective programming problem. *International Journal of Systems Sciences*, **24**, 789–796, 1993.

[3] R. Caballero, E. Cerdá, M. M. Munoz, L. Rey, and I. M. Stancu-Minasian. Efficient solution concepts and their relations in stochastic multi-objective programming. *Journal of Optimization Theory and Applications*, **110**, 53–74, 2001.

[4] D. R. Carino, T. Kent, D. H. Meyers, C. Stacy, M. Sylvanus, A. L. Turner, K. Watanabe, and W. T. Ziemba. The Russell-Yasuda Kasai Model: an asset liability model for a Japanese insurance company using multistage stochastic programming. *Interfaces*, **24**, 29–49, 1994.

[5] V. Charles. E-model for Transportation Problem of Linear Stochastic Fractional Programming. Optimization online, 2007. Available at http://www.optimization-online.org/DB_FILE/2007/03/1607.pdf

[6] V. Charles, S. I. Ansari, and M. M. Khalid. Multi-Objective Stochastic Linear Programming with General form of Distributions. Optimization online, 2009. Available athttp://www.optimization-online.org/DB_FILE/2009/11/2448.pdf

[7] V. Charles and D. Dutta. Bi-weighted multi-objective stochastic fractional programming problem with mixed constraints. In *Second National Conference on Mathematical and Computational Models,* R. Natarajan and G. Arulmozhi, eds, Allied Publishers, 2003.

[8] V. Charles and D. Dutta. Linear Stochastic Fractional Programming with Sum-of-Probabilistic-Fractional Objective. Optimization online, 2005. Available at http://www.optimization-online.org/DB_FILE/2005/06/1142.pdf

[9] A. P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence.* John Wiley & Sons Ltd, 2005.

[10] M. Fisher, J. Hammond, W. Obermeyer, and A. Raman. Conguring a supply chain to reduce the cost of demand uncertainty. *Production and Operations Management*, **6**, 211–225, 1997.

[11] A. Goicoechea, D. R. Hansen, and L. Duckstein. *Multi-objective Decision Analysis with Engineering and Business Application*. John Wiley, 1982.

[12] J.P. Leclercq. Stochastic programming: an interactive multiple approach. *European Journal of Operations Research*, **10**, 33–41, 1982.

[13] F. H. Murphy, S. Sen, and A. L. Soyster. Electric utility capacity expansion planning with uncertain load forecasts. *AIIE Transaction*, **14**, 52–59, 1982.

[14] M. Pant, R. Thangaraj, and V. P. Singh. Optimization of mechanical design problems using improved differential evolution algorithm. *International Journal of Recent Trends in Engineering*, **1**, 21–25, 2009.

[15] N. P. Sahoo and M. P. Biswal. Computation of Probabilistic linear programming problems involving normal and log-normal random variables with a joint constraint. *International Journal of Computer Mathematics*, **82**, 1323–1338, 2005.

[16] S. Sen. Stochastic programming: computational issues and challenges. In *Encyclopedia of OR/MS*, S. Gass and C. Harris, eds, pp. 1–11, 2001.

[17] S. Sen, R. D. Doverspike and S. Cosares. Network planning with random demand. *Telecommunication Systems*, **3**, 11–30, 1994.

[18] R. Stom. System design by constraint adaptation and differential evolution. *IEEE Transactions on Evolutionary Computation*, **3**, 22–34, 1999.

[19] R. Storn and K. Price. Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical Report.* International Computer Science Institute, 1995.

[20] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal Global Optimization* **11**, 341–359, 1997.

[21] H. Suwarna, M. P. Biswal, and S. B. Sinha. Fuzzy programming approach to multiobjective stochastic linear programming problems. *Fuzzy Sets and Systems*, **88**, 173–181, 1997.

[22] R. Thangaraj, M. Pant and A. Abraham. New mutation schemes for differential evolution algorithm and their application to the optimization of directional overcurrent relay settings. *Applied Mathematics and Computation*, **216**, 532–544, 2010.

Received 19 October 2010