# Solving Multi Objective Stochastic Programming Problems Using Differential Evolution

Radha Thangaraj[1], Millie Pant[2], Pascal Bouvry[1], and Ajith Abraham[3]

[1] Faculty of Science, Technology and Communications, University of Luxembourg
[2] Department of Paper Technology, Indian Institute of Technology Roorkee, India
[3] Machine Intelligent Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, USA
t.radha@ieee.org, millifpt@iitr.ernet.in, pascal.bouvry@uni.lu, ajith.abraham@ieee.org

**Abstract.** Stochastic (or probabilistic) programming is an optimization technique in which the constraints and/or the objective function of an optimization problem contains 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, Differential Evolution (DE) and its two recent variants LDE1 and LDE2 are presented for solving multi objective linear stochastic programming (MOSLP) problems, having several conflicting objectives. The numerical results obtained by DE and its variants are compared with the available results from where it is observed that the DE and its variants significantly improve the quality of solution of the given considered problem in comparison with the quoted results in the literature.

**Keywords:** Differential Evolution, stochastic 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 stochastic programming these numbers follow a probability distribution. Thus we can say that SP is a framework for modeling 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 models are formulated, solved analytically or numerically, and analyzed in order to provide useful information to a decision-maker.

In the recent past, SP has been applied to the problems having multiple, conflicting and non-commensurable objectives where generally there does not exist a single solution which can optimize all the objectives. Several methods for solving Multi-Objective Stochastic Linear Programming (MOSLP) problems and their applications to various fields are available in literature [1] – [7]. Most of the probabilistic models

B.K. Panigrahi et al. (Eds.): SEMCCO 2010, LNCS 6466, pp. 54–61, 2010.

assume normal distribution for model coefficients. Sahoo and Biswal [8] presented some deterministic equivalents for the probabilistic problem involving normal and log-normal random variables for joint constraints. Charles et al. [9] 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 followed the models proposed by Charles et al. [9] and have solved them using Differential Evolution algorithm.

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

## 2 Differential Evolution Algorithms

### 2.1 Classical Differential Evolution (DE)

Differential Evolution [10] is a population based metaheuristics that has been consistently ranked as one of the best search algorithm for solving benchmark as well as real life problems in several case studies.

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* [10] which is perhaps the most frequently used version of DE.

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

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

where $r_1, r_2, r_3 \in \{1,2,....,NP\}$ are randomly chosen integers, must be different from each other and also different from the running index *i*. *F* (>0) is a scaling factor which controls the amplification of the differential evolution $(x_{r_2,g} - x_{r_3,g})$. In order to increase the diversity of the perturbed parameter vectors, crossover is introduced. The parent vector is mixed with the mutated vector to produce a trial vector $u_{ji,g+1}$ ,

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

where j = 1, 2,......, D; $rand_j \in [0,1]$ ; CR is the crossover constant takes values in the range [0, 1] and $j_{rand} \in (1,2,.....,D)$ is the randomly chosen index.

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:

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

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 from the tournament selection to the population of 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 Laplace Differential Evolution (LDE)

The LDE algorithms are proposed by Thangaraj et al. [11]. These algorithms differ from the classical DE in the mutation phase in a twofold manner. These schemes make use the absolute weighted difference between the two vector points in place of the usual vector difference as in classical DE and secondly, in LDE schemes amplification factor, F (of the usual DE), is replaced by L, a random variable following Laplace distribution.

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

#### 2.2.1 LDE1 Scheme

$$v_{i,g+1} = x_{best,g} + \mathrm{L}^* \mid x_{r_1,g} - x_{r_2,g} \mid \tag{4}$$

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

#### 2.2.2 LDE2 Scheme

$$\begin{aligned} &If\,(U(0,1) < 0.5)\; then \;\; v_{i,g+1} = x_{best,g} + \mathrm{L}^* \mid x_{r_1,g} - x_{r_2,g} \mid \\ &Else \;\; v_{i,g+1} = x_{r_1,g} + F^*(x_{r_2,g} - x_{r_3,g}) \end{aligned}$$

In LDE2 scheme, mutant vector using equation (4) 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 Eqn. (1) is applied.

Both the modified versions, LDE1 and LDE2 have reportedly given good performances for solving benchmark as well as real life problems [11].

## 3 Problem Definition

Mathematical model of a constrained MOSLP may be given as [9]:

Maximize $z_k = \sum_{j=1}^{n} c_j^k x_j \,,\; k = 1,2,...,K$

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

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 [9]. In the present study, we have considered the two test problems which are used in [9]. These problems are multi-objective stochastic linear programming problems involving random variables following different distributions.

**Test problem 1: MOSLP1:**

$Maximize\ z_1 = 5x_1 + 6x_2 + 3x_3$, $Maximize\ z_2 = 6x_1 + 3x_2 + 5x_3$,

$Maximize\ z_3 = 2x_1 + 5x_2 + 8x_3$

Subject to

$P(3x_1 + 2x_2 + 2x_3 \le b_1) \ge 0.90$, $P(2x_1 + 8x_2 + 5x_3 \le b_2) \ge 0.98$

$P(5x_1 + 3x_2 + 2x_3 \le b_3) \ge 0.95$, $P(0.5x_1 + 0.5x_2 + 0.25x_3 \le b_4) \ge 0.90$

$P(8x_1 + 3x_2 + 4x_3 \le b_5) \ge 0.99$, $x_1, x_2, x_3 \ge 0$

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:

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

Subject to

$3x_1 + 2x_2 + 2x_3 \le 6.3096$, $2x_1 + 8x_2 + 5x_3 \le 8.0812$, $5x_1 + 3x_2 + 2x_3 \le 4.7115$,

$0.5x_1 + 0.5x_2 + 0.25x_3 \le 0.9379$, $8x_1 + 3x_2 + 4x_3 \le 10.0321$, $\lambda_1 + \lambda_2 + \lambda_3 = 1$

$x_1, x_2, x_3, \lambda_1, \lambda_2, \lambda_3 \ge 0$

**Test problem 2: MOSLP2:**

$Maximize\ z_1 = 3x_1 + 8x_2 + 5x_3$, $Maximize\ z_2 = 7x_1 + 4x_2 + 3x_3$

$Maximize\ z_3 = 6x_1 + 7x_2 + 10.5x_3$

Subject to

$P(5x_1 + 4x_2 + 2x_3 \le b_1) \ge 0.95$, $P(7x_1 + 3x_2 + x_3 \le b_2) \ge 0.95$

$P(2x_1 + 7x_2 + 3x_3 \le b_3) \ge 0.95$, $P(2x_1 + 3x_2 + 2.5x_3 \le b_4) \ge 0.95$

$P(5x_1 + 2x_2 + 1.5x_3 \le b_5) \ge 0.95$, $x_1, x_2, x_3 \ge 0$

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:

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

Subject to

$$\left[\frac{y_1^2}{9}\right]\left[\frac{y_2^2 - 100}{y_2^2}\right]\left[\frac{y_3 - 5}{10}\right]\left[\frac{e^{2y_4} - 1}{e^{2y_4}}\right]\left[\frac{3y_5^2}{1 + 3y_5^2}\right] \ge 0.95$$

$$5x_1 + 4x_2 + 2x_3 = y_1,\ 7x_1 + 3x_2 + x_3 = y_2,\ 2x_1 + 7x_2 + 3x_3 = y_3,$$
$$2x_1 + 3x_2 + 2.5x_3 = y_4,\ 5x_1 + 2x_2 + 1.5x_3 = y_5,\ \lambda_1 + \lambda_2 + \lambda_3 = 1$$
$$x_1, x_2, x_3, y_1, y_2, y_3, y_4, y_5, \lambda_1, \lambda_2, \lambda_3 \geq 0.$$

# 4 Experimental Settings and Numerical Results

## 4.1 Parameter Settings

DE has three main control parameters; population size, crossover rate Cr and Scaling factor F which 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 the maximum number of generations is reached (assumed 1000 generations). Constraints are handled according to the approach based on repair methods suggested in [12]. 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 basic DE and LDE versions are also compared with previously quoted results [9].

## 4.2 Numerical Results

We have considered four test cases in each of the test problems. Since, $\lambda_1 + \lambda_2 + \lambda_3 = 1$, one of $\lambda_i$, $i$ = 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) $\lambda_1$, $\lambda_2$, and $\lambda_3$ are dependent variables.

The numerical results of the given two test problems MOSLP1 and MOSLP2 are recorded in Tables 1 and 2 respectively. The best solution obtained by DE and LDE algorithms for MOSLP1 in terms of optimal decision variable values and objective function value are given in Table 1. For the test case (i), the performance of LDE1 is better than all the other algorithms. For the remaining 3 test cases, LDE2 performs better than other compared algorithms. If we compare the LDE algorithms with classical DE algorithm then from the numerical results we can see that LDE algorithms are superior with classical DE algorithm. There is an improvement of 52% in objective function value when the problem is solved by LDE2 in comparison with the quoted result [9], where the problem is solved by Genetic Algorithm. The results of test problem MOSLP2 are given in Table 2. 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 DE and LDE algorithms in terms of objective function value.

**Table 1.** Results of MOSLP1

| | DE | LDE1 | LDE2 | GA [9] |
|---|---|---|---|---|
| **$\lambda_1 = W, \lambda_1 = \lambda_2 = (1-W)/2, 0 \le W \le 1$** | | | | |
| z | 10.9905 | **10.997** | 10.996 | --NA-- |
| $x_1$ | 0.349128 | 0.351905 | 0.35171 | |
| $x_2$ | 0 | 0 | 0 | |
| $x_3$ | 1.47618 | 1.47538 | 1.47539 | |
| **$\lambda_2 = W, \lambda_1 = \lambda_3 = (1-W)/2, 0 \le W \le 1$** | | | | |
| z | 9.48974 | **9.48975** | **9.48975** | --NA-- |
| $x_1$ | 0.35214 | 0.35215 | 0.352142 | |
| $x_2$ | 0 | 0 | 0 | |
| $x_3$ | 1.47538 | 1.47537 | 1.47538 | |
| **$\lambda_3 = W, \lambda_1 = \lambda_2 = (1-W)/2, 0 \le W \le 1$** | | | | |
| z | 12.9277 | 12.9288 | **12.9292** | --NA-- |
| $x_1$ | 0 | 0 | 0 | |
| $x_2$ | 0 | 0 | 0 | |
| $x_3$ | 1.61611 | 1.61612 | 1.61617 | |
| **Problem described as in [9]** | | | | |
| z | 9.48978 | 11.3988 | **12.9299** | 8.5089 |
| $x_1$ | 0.352147 | 0.334378 | 0 | 0.3727 |
| $x_2$ | 2.12479e-007 | 0.00514505 | 0 | 0.2319 |
| $x_3$ | 1.47538 | 1.47426 | 1.61624 | 1.0761 |

**Table 2.** Results of MOSLP2

| | DE | LDE1 | LDE2 | GA [9] |
|---|---|---|---|---|
| **$\lambda_1 = W, \lambda_1 = \lambda_2 = (1-W)/2, 0 \le W \le 1$** | | | | |
| z | 5.5452 | 6.3844 | **6.86328** | --NA-- |
| $x_1$ | 0.170342 | 0.275175 | 0.297729 | |
| $x_2$ | 0.0367932 | 0.0654974 | 0.00485206 | |
| $x_3$ | 0.759151 | 0.627495 | 0.726168 | |
| $y_1$ | 2.5158 | 2.89285 | 2.96039 | |
| $y_2$ | 2.06291 | 2.7502 | 2.82483 | |
| $y_3$ | 2.862 | 2.89131 | 2.80793 | |
| $y_4$ | 2.36484 | 2.31558 | 2.42544 | |
| $y_5$ | 2.06754 | 2.44811 | 2.5876 | |
| **$\lambda_2 = W, \lambda_1 = \lambda_3 = (1-W)/2, 0 \le W \le 1$** | | | | |
| z | 5.3215 | 7.01255 | **7.72732** | --NA-- |
| $x_1$ | 0.170342 | 0.12258 | 0 | |
| $x_2$ | 0.0367932 | 0.0575791 | 0.00166162 | |
| $x_3$ | 0.759151 | 0.777962 | 0.995503 | |
| $y_1$ | 2.5158 | 2.39914 | 1.99765 | |
| $y_2$ | 2.06291 | 1.80875 | 1.00048 | |
| $y_3$ | 2.862 | 2.98209 | 2.99814 | |
| $y_4$ | 2.36484 | 2.36281 | 2.49374 | |
| $y_5$ | 2.06754 | 1.895 | 1.49658 | |
| **$\lambda_3 = W, \lambda_1 = \lambda_2 = (1-W)/2, 0 \le W \le 1$** | | | | |
| z | 6.60213 | 9.3271 | **10.4638** | --NA-- |
| $x_1$ | 0.170342 | 0.126015 | 0 | |
| $x_2$ | 0.0367932 | 0 | 0.00166304 | |
| $x_3$ | 0.759151 | 0.816303 | 0.995504 | |
| $y_1$ | 2.5158 | 2.26268 | 1.99765 | |
| $y_2$ | 2.06291 | 1.69841 | 1.00049 | |
| $y_3$ | 2.862 | 2.70093 | 2.99815 | |

**Table 2.** (*continued*)

| | | | | |
|---|---|---|---|---|
| $y_4$ | 2.36484 | 2.29278 | 2.49374 | |
| $y_5$ | 2.06754 | 1.85453 | 1.49659 | |
| **Problem described as in [9]** | | | | |
| z | 6.87235 | 7.13425 | **7.73912** | 3.2081 |
| $x_1$ | 2.65138e-006 | 0.000944931 | 0.000308158 | 0.1939 |
| $x_2$ | 0.000127494 | 0.061029 | 0.127573 | 0.2810 |
| $x_3$ | 0.664552 | 0.738963 | 0.688939 | 0.1968 |
| $y_1$ | 1.32963 | 1.72678 | 1.88971 | 2.4872 |
| $y_2$ | 0.664947 | 0.928675 | 1.07383 | 2.3971 |
| $y_3$ | 1.99454 | 2.64598 | 2.96046 | 2.9454 |
| $y_4$ | 1.66177 | 2.03239 | 2.10569 | 1.7229 |
| $y_5$ | 0.9971 | 1.0 | 1.0 | 1.8267 |

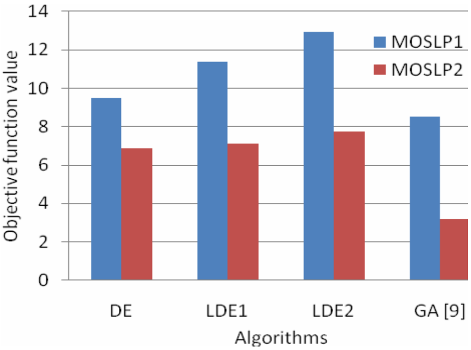

**Fig. 1.** Performance of DE and LDE algorithms in terms of objective function value

# 5 Conclusion

The Stochastic Programming 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 DE and two of its recent variants LDE1 and LDE2 are used to solve two constrained multiobjective stochastic linear programming problems. Four test cases were considered with respect to the weighing factors and the results were produced in terms of objective function value and decision variable values. From the experimental results it was observed that the DE algorithm and its variants significantly improve the quality of solution of the considered problems in comparison with the quoted results in the literature. As expected the modified versions LDE1 and LDE2 performed better than the basic version of DE because of the presence of the Laplace mutation operator. In conclusion we can say that DE's present an attractive option for solving stochastic programming problems.

## Acknowledgement

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

## References

1. Abdelaziz, F.B., Aouni, B., Rimeh, F.E.: Multi-objective programming for portfolio selection. European Journal Operational Research 177(3), 1811–1823 (2007)
2. Baba, N., Morimoto, A.: Stochastic approximations methods for solving the stochastic multi-objective programming problem. International Journal of Systems Sciences 24, 789–796 (1993)
3. Caballero, R., Cerdá, E., Munoz, M.M., Rey, L., Stancu-Minasian, I.M.: Efficient solution concepts and their relations in stochastic multi-objective programming. Journal of Optimization Theory and Applications 110, 53–74 (2001)
4. Charles, V., Dutta, D.: Bi-weighted multi-objective stochastic fractional programming problem with mixed constraints. In: Natarajan, R., Arulmozhi, G. (eds.) Second National Conference on Mathematical and Computational Models. Allied Publishers, Chennai (2003)
5. Goicoechea, A., Hansen, D.R., Duckstein, L.: Multi-objective Decision Analysis with Engineering and Business Application. John Wiley, New York (1982)
6. Leclercq, J.P.: Stochastic Programming: An Interactive Multiple Approach. European Journal of Operations Research 10, 33–41 (1982)
7. Suwarna, H., Biswal, M.P., Sinha, S.B.: Fuzzy programming approach to multiobjective stochastic linear programming problems. Fuzzy Sets and Systems 88, 173–181 (1997)
8. Sahoo, N.P., Biswal, M.P.: Computation of Probabilistic linear programming problems involving normal and log-normal random variables with a joint constraint. International Journal of Computer Mathematics 82(11), 1323–1338 (2005)
9. Charles, V., Ansari, S.I., Khalid, M.M.: Multi-Objective Stochastic Linear Programming with General form of Distributions, http://www.optimization-online.org/DB_FILE/2009/11/2448.pdf
10. Storn, R., Price, K.: Differential Evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report, International Computer Science Institute, Berkley (1995)
11. Thangaraj, R., Pant, M., Abraham, A.: New Mutation Schemes for Differential Evolution Algorithm and their application to the Optimization of Directional Overcurrent Relay Settings. Applied Mathematics and Computation 216(2), 532–544 (2010)
12. Pant, M., Thangaraj, R., Singh, V.P.: Optimization of Mechanical Design Problems using Improved Differential Evolution Algorithm. Int. Journal of Recent Trends in Engineering 1(5), 21–25 (2009)