

Cairo University

Egyptian Informatics Journal

[www.elsevier.com/locate/eij](http://www.elsevier.com/locate/eij) [www.sciencedirect.com](http://www.sciencedirect.com/science/journal/11108665)



FULL-LENGTH ARTICLE

Solving stochastic programming problems using new approach to Differential Evolution algorithm



Ali Wagdy Mohamed

*Operations Research Department, Institute of Statistical Studies and Research, Cairo University, Giza 12613, Egypt*

Received 2 October 2015; revised 24 August 2016; accepted 27 September 2016

Available online 27 October 2016

Abstract This paper presents a new approach to Differential Evolution algorithm for solving stochastic programming problems, named DESP. The proposed algorithm introduces a new trian- gular mutation rule based on the convex combination vector of the triangle and the difference vec- tor between the best and the worst individuals among the three randomly selected vectors. The proposed novel approach to mutation operator is shown to enhance the global and local search capabilities and to increase the convergence speed of the new algorithm compared with conven- tional DE. DESP uses Deb’s constraint handling technique based on feasibility and the sum of con- straint violations without any additional parameters. Besides, a new dynamic tolerance technique to handle equality constraints is also adopted. Two models of stochastic programming (SP) problems are considered: Linear Stochastic Fractional Programming Problems and Multi-objective Stochas- tic Linear Programming Problems. The comparison results between the DESP and basic DE, basic particle swarm optimization (PSO), Genetic Algorithm (GA) and the available results from where it is indicated that the proposed DESP algorithm is competitive with, and in some cases superior to, other algorithms in terms of final solution quality, efficiency and robustness of the considered prob- lems in comparison with the quoted results in the literature.

© 2016 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information,

KEYWORDS

Differential evolution; Stochastic programming; Fractional programming; Multi-objective programming

Cairo University. This is an open access article under the CC BY-NC-ND license [(http://creativecommons.](http://creativecommons/)

org/licenses/by-nc-nd/4.0/).

1. Introduction

Stochastic or probabilistic programming (SP) deals with situa- tions where some or all of the parameters of the optimization

E-mail address: [aliwagdy@gmail.com](mailto:aliwagdy@gmail.com)

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

**Production and hosting by Elsevier**

problem are described by random or probabilistic variables rather than by deterministic quantities [[1]](#_bookmark16). The mathematical models of these problems may follow any particular probabil- ity distribution for model coefficients [[2]](#_bookmark17). The main objective is to find the optimal value for model parameters influenced by random event. The basic idea used in stochastic programming is to convert the stochastic problem into an equivalent deter- ministic problem which can be solved by using an appropriate classical and/or modern numerical technique. SP is widely used in many real-world decision making problems of management science, engineering, and technology. Also, it has been applied to various areas such as finance [[3]](#_bookmark18), manufacturing product

<http://dx.doi.org/10.1016/j.eij.2016.09.002>

1110-8665 © 2016 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University. This is an open access article under the CC BY-NC-ND license [(http://creativecommons.org/licenses/by-nc-nd/4.0/).](http://creativecommons.org/licenses/by-nc-nd/4.0/))

and capacity planning [[4]](#_bookmark19), electrical generation capacity plan- ning [[5]](#_bookmark20), supply chain management [[6]](#_bookmark21), water resource alloca- tion modeling [[7]](#_bookmark22), portfolio selection [[8]](#_bookmark23), project selection [[9]](#_bookmark24),

transportation [[10]](#_bookmark25), Telecommunications [[11]](#_bookmark26), energy planning

[[12]](#_bookmark28), healthcare management [[13]](#_bookmark29) and marketing [[14]](#_bookmark30). A con- strained 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 prob- lem data is random in nature with nonnegative constraints on the variables. In addition, some of the constraints may be deterministic. Thus, 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 [[15]](#_bookmark30). There are many algorithms to solve LSFPP such as [[16–19]](#_bookmark30). Moreover, 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 [[20]](#_bookmark30). Various algorithms to solve Multiobjective Stochastic Linear Programming (MOSLP) problems have been discussed in the literature [[20–23]](#_bookmark30). On the other hand, over the last three dec- ades, Evolutionary Algorithms (EAs) have been developed to solve nonlinear, high-dimensional and complex computational optimization problems. Virtually, EAs have been originally proposed to overcome the challenges of global optimization problems such as nonlinearity, non-convexity, non- continuity, non-differentiability, and/or multimodality, while

equality constraints is also adopted, (3) the crossover rate in

[[27]](#_bookmark31) is given by a dynamic nonlinear increased probability scheme, but in this work, the crossover rate is fixed 0.95. (4) The triangular mutation rule is only used in this work, but in the previous work [[27]](#_bookmark31), the triangular mutation strategy is embedded into the DE algorithm and combined with the basic mutation strategy DE/rand/1/bin through a nonlinear decreas- ing probability rule. (5) In previous work [[27]](#_bookmark31) a restart mech- anism based on Random Mutation and modified BGA mutation is used to avoid stagnation or premature conver- gence, whereas this work does not. The remainder of this paper is organized as follows: Section [2](#_bookmark1) presents the problem of our interest and the Deb’s procedure for handling constraints. In Section [3](#_bookmark2), the standard DE algorithm is introduced with a review of its operators and parameters. Next, in Section [4](#_bookmark4), the proposed DESP algorithm is described. The problem defi- nitions are given in Section [5](#_bookmark8). In Section [6](#_bookmark11), the effectiveness of the proposed method, the experimental settings and numerical results are discussed. Finally, the conclusions and future works are drawn in Section [7](#_bookmark27).

1. Problem statement

In general, constrained optimization problem can be expressed as follows (without loss of generality minimization is consid- ered here):

→ →

traditional numerical optimization techniques had difficulties

ffi*n* (1)

with these problems. Differential Evolution (DE) is a stochas- tic population-based search method, proposed by Storn and Price [[24]](#_bookmark34). DE is considered the most recent EAs for solving real-parameter optimization problems [[25]](#_bookmark35). 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 [[26]](#_bookmark37). In this research, two models of stochas- tic programming (SP) problems are considered: Linear Stochastic Fractional Programming Problems and Multi- Objective Stochastic Linear Programming Problems. For LSFPP and MOSLP, the models proposed by Charles and

Minimize *f*(*x*); *x* = (*x*1; *x*2; ... ; *xn*)∈

Subject to:

→

*gj*(*x*) 6 0; *j* = 1; ... ; *q* (2)

→

*hj*(*x*)= 0; *j* = *q* + 1; .. . ; *m* (3)

→

where *x* ∈ X C *S*, X is the feasible region, and *S* is an

*n* -dimensional rectangular space in ffi*n* defined by the paramet- ric constraints *li* 6 *xi* 6 *ui*; 1 6 *i* 6 *n* where *li* and *ui* are lower and upper bounds for a decision variable *xi*, respectively.

→

For an inequality constraint that satisfies *gj* (*x*) 6 0

→ →

Dutta [[19]](#_bookmark30) and by Charles et al. [[20]](#_bookmark30), respectively, are followed.

The deterministic equivalent models of these two classes of stochastic programming models are solved using a new approach to Differential Evolution algorithm, named DESP. The proposed algorithm introduces a new triangular mutation

(*j* ∈ 1; .. . ; *q*) at any point *x* ∈ X, we say it is active at *x*. All

→

equality constraints *hj*(*x*)= 0; (*j* = *q* + 1; ... ; *m*) are consid-

ered active at all points of X. Most constraint-handling approaches used with EAs tend to deal only with inequality constraints. Therefore equality constraints are transformed

rule based on the convex combination vector of the triangle →

into inequality constraints of the form *hj*(*x*) — *e* 6 0 where

and the difference vector between the best and the worst indi-

viduals among the three randomly selected vectors. The pro- posed novel approach to mutation operator is shown to enhance the global and local search capabilities and to increase the convergence speed of the new algorithm compared with conventional DE. DESP uses Deb’s constraint handling tech- nique based on feasibility and the sum of constraint violations without any additional parameters. Besides, a new dynamic tolerance technique to handle equality constraints is also adopted. The results obtained by DESP algorithms are com- pared with basic versions of DE and PSO and also with the results quoted in the literature [[19,20]](#_bookmark30). It is worth noting that although this work is an extension and modification of our previous work in [[27]](#_bookmark31), there are significant differences as fol- lows: (1) previous work in [[27]](#_bookmark31) is proposed for unconstrained problems, whereas this work is proposed for constrained prob- lems. Hence, (2) a new dynamic tolerance technique to handle

*e* is the tolerance allowed (a very small value) [[28]](#_bookmark31). In order

to handle constraints, we use Deb’s constraint handling proce- dure. Deb [[29]](#_bookmark31) proposed a new efficient feasibility-based rule to handle constraint for genetic algorithm where pair-wise solu- tions are compared using the following criteria:

* Between two feasible solutions, the one with the highest fit- ness values wins.
* If one solution is feasible and the other one is infeasible, the feasible solution wins.
* If both solutions are infeasible, the one with the lowest sum of constraint violation is preferred.

As a result, Deb [[29]](#_bookmark31) has introduced the superiority of fea- sible solutions selection procedure based on the idea that any individual in a constrained search space must first comply with

the constraints and then with the objective function. Practi- cally, Deb’s selection criterion does not need to fine-tune any parameter. Typically, from the problem formulation above, there are *m* constraints and hence the amount of constraint violation for an individual is represented by a vector of *m* dimensions. Using a tolerance (*e*) allowed for equality con-

straints, the constraint violation of a decision vector or an indi-

*3.3. Mutation*

At generation *G*, for each target vector *xG*, a mutant vector

*i*

*vG*+1 is generated according to the following:

*i*

*vG*+1 = *xG* + *F* \* *xG* — *xG* ; *r*1 – *r*2 – *r*3 – *i* (8)

*i*

*r*1

*r*2

*r*3

→

vidual *x* on the *j*th constraint is calculated by

with randomly chosen indices *r*1; *r*2; *r*3 ∈ {1; 2; ... ; *NP*}. *F* is a

real number to control the amplification of the difference vec-

*F* is in [0, 2]. If a component of a mutant vector violates search

space, then the value of this component is generated a new

→ ( max(0; *gj* (→*x* )); *j* = 1; .. . ; *q*

tor *xG* — *xG* . According to Storn and Price [[34]](#_bookmark31), the range of

*cvj*( *x* )=

→

*j*

max(0; *h* ( *x* ) — *e*); *j* = *q* + 1; .. . ; *m*

(4)

*r*2 *r*3

If a decision vector or an individual *x* satisfies the *jth* con-

→

→

straint, *cj*(*x*) is set to zero. As discussed [[30]](#_bookmark31), in order to con-

sider all the constraints at the same time or to treat each constraint equally, each constraint violation is then normalized by dividing it by the largest violation of that constraint in the population. Thus, the maximum violation of each constraint in the population is given by

using [(7)](#_bookmark5) or new other repair method.

*3.4. Crossover*

The binomial crossover, the target vector is mixed with the mutated vector, using the following scheme, to yield the trial vector *uG*+1.

*i*

*c j* = max

max *x*→∈*s*

→

*cj*(*x*)

(5)

*G*+1

*u* =

*ij*

*G*+1

*ij*

( *v* ; *rand*(*j*) 6 *CR* or *j* = *randn*(*i*);

*xG*; *rand*(*j*) > *CR* and *j* – *randn*(*i*);

(9)

These maximum constraint violation values are used to normalize each constraint violation. The normalized constraint violations are added together to produce a scalar constraint

→

violation *c*(*x*) for that individual which takes a value between

0 and 1

*ij*

where *j* = 1; 2; .. . ; *D*, *rand*(*j*)∈ [0; 1] is the *j*th evaluation of a uniform random generator number. *CR* ∈ [0; 1] is the cross- over rate, *randn*(*i*)∈ {1; 2; ... ; *D*} is a randomly chosen index

which ensures that *uG*+1 gets at least one element from *vG*+1;

*i* *i*

otherwise, no new parent vector would be produced and the

*m* →

→ 1 X *cj*(*x*)

*c*

*c*(*x*)= *m*

(6)

*j*=1

*j*

max

population would not alter.

*3.5. Selection*

1. Methodology
   1. *Differential Evolution (DE)*

DE adapts a greedy selection strategy. If and only if the trial vector *uG*+1 yields as good as or a better fitness function value

than *xG*, then *uG*+1 is set to *xG*+1. Otherwise, the old vector *xG* is

*i*

*i i i* *i*

This section provides a brief summary of the basic Differential Evolution (DE) algorithm. In simple DE, generally known as

DE/rand/1/bin [[31,32]](#_bookmark31), an initial random population consists

retained. The selection scheme is as follows (for a minimization

problem):

( *uG*+1; *f* *uG*+1 6 *f*(*xG*)

*i*

*xG*+1 =

*i*

*i*

*i*

(10)

→

of *NP* vectors *Xi*, 6*i* = 1; 2; ... ; *NP*, and is randomly generated

according to a uniform distribution within the lower and upper boundaries *xL*; *xU* . After initialization, these individuals are

*j*

*j*

evolved by DE operators (mutation and crossover) to generate a trial vector. A comparison between the parent and its trial vector is then done to select the vector which should survive to the next generation [[33]](#_bookmark31). DE steps are discussed below:

* 1. *Initialization*

In order to establish a starting point for the optimization pro- cess, an initial population must be created. Typically, each decision variable in every vector of the initial population is assigned a randomly chosen value from the boundary constraints:

*x*0 = *xL* + *randj* · *xU* — *xL* (7)

*ij*

*j*

*j*

*j*

where *randj* denotes a uniformly distributed number between [0, 1], generating a new value for each decision variable.

*i xG*; otherwise

A detailed description of standard DE algorithm is given in [Fig. 1](#_bookmark6).

1. DESP algorithm

All evolutionary algorithms, including DE, are stochastic population-based search methods. Accordingly, there is no guarantee that the global optimal solution will be reached con- sistently. Furthermore, they are not originally designed to solve constrained optimization problems. Nonetheless, adjust- ing control parameters such as the scaling factor, the crossover rate and the population size, alongside with developing an appropriate mutation scheme and coupling with suitable and effective constraint-handling techniques can considerably improve the search capability of DE algorithms. Moreover, the possibility of achieving promising and successful results in complex numerical and engineering constrained optimiza-

tion problems increases. Therefore, in this paper, three modifi-

cations are introduced in order to significantly enhance the overall performance of the standard DE algorithm.

|  |  |
| --- | --- |
|  |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Figure 1 Description of standard DE algorithm. rand [0, 1) is a function that returns a real number between 0 and 1. randint (min, max) is a function that returns an integer number between min and max. *NP*, *GEN*, *CR* and *F* are user-defined parameters. *D* is the dimensionality of the problem.

* 1. *Triangular mutation*

Practical experience through many developed evolutionary algorithms and experimental investigation prove that the suc- cess of population-based search algorithms is based on main- taining a balance between two contradictory aspects: exploration and exploitation [[35]](#_bookmark31). Moreover, the mutation

the local search tendency, to improve the global exploration ability and to accelerate the convergence of standard DE tech- nique, a new triangular mutation rule is proposed based on the convex combination vector of the triangle and the difference vector between the best and the worst individuals among the three randomly selected vectors. The modified mutation scheme is as follows:

scheme plays a vital role in the DE search capability and the

*mG*+1 = *x*¯*G* + 2*F* · *xG*

— *xG*

(11)

convergence rate. However, even though the DE algorithm

has good global exploration ability, it suffers from weak local

*i*;*j*

*c*;*j*

*best*;*j*

*worst*;*j*

exploitation ability and its convergence velocity is still too low

where *x*¯*G* is a convex combination vector of the triangle and

as the region of the optimal solution is reached [[36]](#_bookmark32). Clearly,

*G*

*best*

*x*

*G*

*worst*

*c*

and *x*

are the best and the worst individuals among

from the mutation Eq. [(8)](#_bookmark3), it can be observed that three vectors are randomly chosen for mutation and the base vector is then randomly selected among the three. Consequently, the basic mutation strategy DE/rand/1/bin is able to maintain popula- tion diversity and global search capability, but it slows down

the three randomly selected vectors, respectively. *F* is a uni-

form random variables, *u*(0; 1) returns a real number between

0 and 1 with uniform random probability distribution and *G* is the current generation number. This modification is intended

to replace the random base vector *xG* in Eq. [(8)](#_bookmark3) by the convex

*r*

1

*c*

the convergence of DE algorithms. Hence, in order to enhance

combination vector *x*¯*G*

and the remaining two vectors are

replaced by the best and worst vectors of the three randomly selected vectors to yield the difference vector. The mutation operation in Eq. [(11)](#_bookmark7) is originally performed according to the following mutation equation:

*mG*+1 = *x*¯*G* + *F* · *xG* — *xG* + *F* · *xG* — *xG*

+ *F* · *x*

— *x*

*i*,*j*

*c*,*j*

*best*,*j*

*better*,*j*

*better*,*j*

*worst*,*j*

region of each *x*¯*G* in the direction of *xG* — *xG* for each mutated vector. In a nutshell, it concentrates the exploitation of some sub-regions of the search space. Thus, it has better local search tendency so it accelerates the convergence speed of the proposed algorithm. Besides, the global exploration

where *xG*

*best*

*G*

ability of the algorithm is significantly enhanced by forming

*best*,*j*

*G*

, *x*

and *x*

*better*

*G*

*worst*,*j*

*G*

*worst*

(12)

are the tournament best, better and

*c*

*best*

*worst*

many different sizes and shapes of triangles in the feasible region through the optimization process. Thus, the proposed directed mutation balances both global exploration capability

worst three randomly selected vectors, respectively. The con- vex combination vector *x*¯*G* of the triangle is a linear combina- tion of the three randomly selected vectors and is defined as follows:

*c*

and local exploitation tendency.

* 1. *Modification of scaling factor*

*x*¯*G* = *w*1 · *xG*

*c best*

*G*

*better*

+ *w*2 · *x*

*G*

*worst*

+ *w*3 · *x*

(13)

In the mutation Eq. [(8)](#_bookmark3), the constant of differentiation *F* is a scaling factor of the difference vector. It is an important

where the real weights *w* satisfy *w* P 0 and 3 *w* = 1. Where the real weights *wi* are given by *wi* = *p* /P3 *p* ,

*i*=1

*i*

*i*=1 *i*

DE algorithm in [[30]](#_bookmark31), the constant of differentiation *F* was

*i i* P *i*

parameter that controls the evolving rate of the population. Regarding global unconstrained optimization, in the original

*i* = 1, 2, 3. Where *p*1 = 1, *p*2 = *rand*(0.5, 1) and *p*3 =

*rand*(0, 0.5), *rand*(*a*, *b*) are functions that return a real number

between *a* and *b*, where *a* and *b* are not included. For constrained optimization problems at any generation *g* > 1, the tournament selection of the three randomly selected vectors and assigning weights follow one of the following three scenarios that may exist through generations, Without loss of generality, we only consider minimization problems:

Scenario 1: If the three randomly selected vectors are feasi- ble, then sort them in ascending according to their objective

chosen to be a value in [0, 2]. The value of *F* has a considerable influence on exploration: small values of *F* lead to premature convergence. Besides, it can lead to speed up the convergence, and high values slow down the search [[36]](#_bookmark32). In this paper, for the difference vector in the mutation Eq. [(11)](#_bookmark7), it can be seen that it is a directed difference vector from the worst to the best vectors in the entire population. Hence, *F* must be a positive value in order to bias the search direction for all trial vectors in the same direction. Therefore, *F* is introduced as a uniform random variable within (0, 1]. Instead of keeping *F* constant

function values and assign *w*1, *w*2, *w*3 to *xG*

*best*

*G*

*better*

, *x*

and

during the search process, *F* is set as a random variable for

*G*

*x*

*worst*

, respectively.

each trial vector so as to perturb the random base vector by

Scenario 2: If the three randomly selected vectors are infea-

sible, then sort them in ascending order according to their con- straint violation (*CV*) values and assign *w*1, *w*2, *w*3 to *xG* ,

*best*

different directed weights. Therefore, the new directed muta-

tion resembles the concept of gradient as the difference vector is oriented from the worst to the best vectors [[37,38]](#_bookmark33).

*G*

*x*

*better*

*G*

*worst*

and *x*

, respectively.

Scenario 3: If the three randomly selected vectors are mixed

(feasible and infeasible), then the vectors are sorted by using the three criteria: (a) Sort feasible vectors in front of infeasible solutions, (b) sort feasible solutions according to their objec- tive function values, and (c) sort infeasible solutions according to their constraint violations. Accordingly, assign *w*1, *w*2, *w*3 to

* 1. *Equality constraint approach*

For any feasible region limited by a set of constraints, if one of these constraints is represented by equality, the feasible region becomes a line segment. Furthermore, if all these constraints

are equality constraints, then the feasible region is reduced

*G*

*x*

*best*

*G*

*better*

, *x*

*G*

*worst*

and *x*

, respectively. Obviously, from mutation

to a point i.e. the intersection point of all constraints. There-

Eqs. [(11) and (13)](#_bookmark7), it can be observed that the incorporation

fore, although the equality constraints are transformed into

of the objective function value and constraints violation in →

inequality constraints of the form *hj*(*x*) — *e* 6 0 where *e* is

the mutation scheme has two benefits. Firstly, the perturbation

part of the mutation is formed by the three sides of the triangle

in the direction of the best vector among the three randomly selected vectors. Therefore, the directed perturbation in the proposed mutation resembles the concept of gradient as the difference vector is oriented from the worst to the best vectors [[36]](#_bookmark32). Thus, it is considerably used to explore the landscape of the objective function being optimized in different sub-region around the best vectors within constrained search-space through optimization process. Secondly, the convex combina- tion vector *x*¯*G* is a weighted sum of the three randomly selected vectors where the best vector has the highest contribution. Therefore, *x*¯*G* is extremely affected and biased by the best vec- tor more than the remaining two vectors. Consequently, the global solution can be easily reached if all vectors follow the direction of the best vectors besides they also follow the oppo- site direction of the worst vectors among the randomly selected vectors. Indeed, the new mutation process exploits the nearby

*c*

*c*

the tolerance allowed (a very small value), it still makes the fea-

sible region very small compared with the search space, which exacerbates the optimization process using any evolutionary algorithm to find the feasible solutions. As a result, the larger the tolerance value, the more reliable feasible solutions found [[28,39]](#_bookmark31). In other words, the temporary increase in the feasible space during the initial generations of the search process there will be some feasible individuals from the beginning of the optimization process. Then, by reducing the search space after some generations by decreasing the tolerance value (*e*), the algorithm will improve the previous feasible solutions to fit into the new feasible region and satisfy the new equality con- straints. Hence, in order to deal with equality constraints, the researchers introduce the new approach that dynamically changes the value of (*e*) through generations. The tolerance value (*e*) is adapted and decreased through generations by using the following mathematical expression:

—1 \* (log10 (4)/log10 (10)), 0.00 6 *G*/*GEN* < 0.05

8>

—1 \* (log10 (3)/log10 (10)), 0.05 6 *G*/*GEN* < 0.10

> —1 \* (log10 (2)/log10 (10)), 0.10 6 *G*/*GEN* < 0.15

>< —1 \* (log10

(1)/log10

(10)), 0.15 6 *G*/*GEN* < 0.20

the feasible space is rapidly decreased during the initial 20% of total generations of the search process in constant rate of 5% of total generations. However, to remove the improper

solutions and keep the actual optimal value, the feasible region

is reduced gradually in constant rate of 20% of the total gen-

1, 0.20 6 *G*/*GEN* < 0.40

>

*Factor* =

2, 0.40 6 *G*/*GEN* < 0.60

3, 0.60 6 *G*/*GEN* < 0.80

>: 4, 0.80 6 *G*/*GEN* 6 1.00

(14)

where *e*(*t*)= 10—*Factor*, *G* is the current generation number, *GEN* is the maximum number of generations. As can be easily derived from the above expression, as *G*/*GEN* increases toward 1, the tolerance value *e*(*t*) decreases from 4 to reach

10—(4) and stays fixed in the later generations. In other words, as *G*/*GEN* increases, the widening feasible region considerably shrinks to the true feasible region. Indeed, as number of equal-

ity constraints increases, the initial tolerance has also a signif- icant impact in creating the suitable initial feasible region that can attract too many infeasible points which can be improved through generations to be feasible. Moreover, as previously discussed, it can be observed that, the temporary increase in

erations during the remaining generations. The description of DESP is presented in [Fig. 2](#_bookmark9).

1. Problem definition

This section is divided into two subsections; in Section [1](#_bookmark0), the problem formulation of LSFPPs is given and the general model of the multi-objective SP problems with two test exam- ples is given in Section [2](#_bookmark1).

* 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 vari- ables. Additionally, some of the constraints may be determin- istic [[40]](#_bookmark36). The LSFP framework attempts to model uncertainty

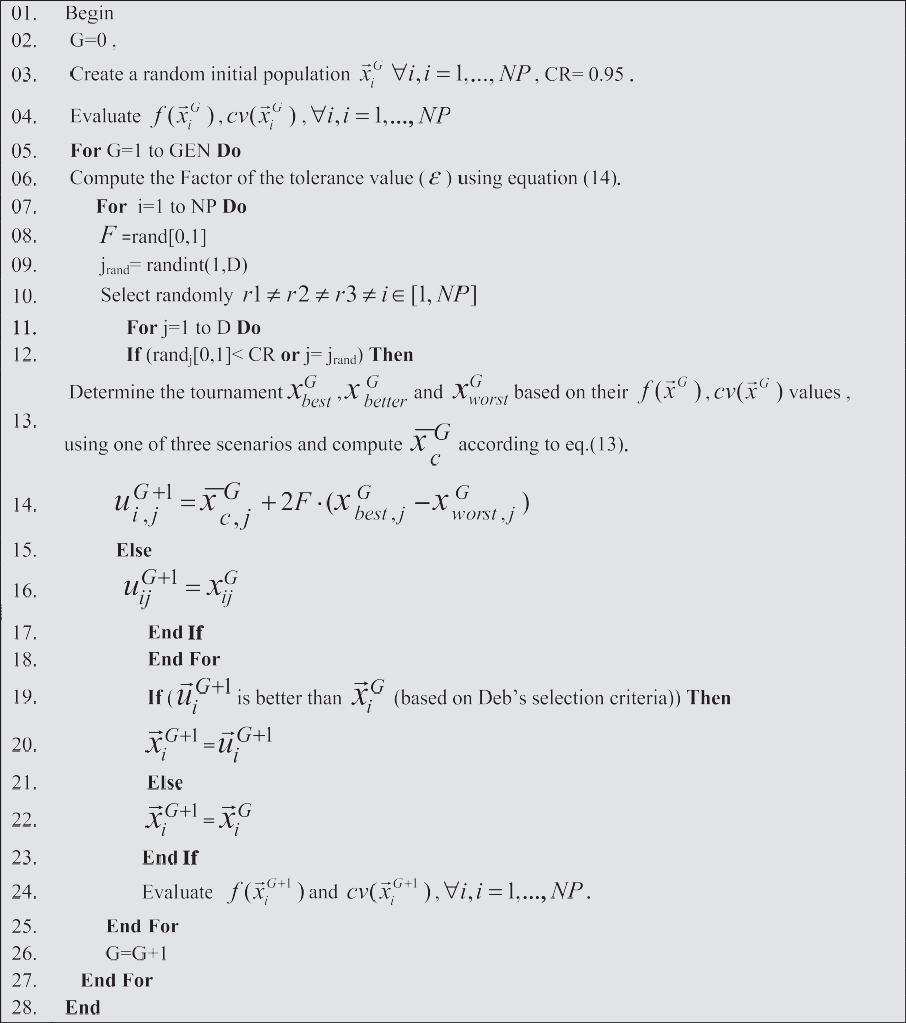


Figure 2 Description of DESP algorithm.

in the data by assuming that the input or a part there of is spec- ified by a probability distribution, rather than being determin- istic. 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.

The mathematical model of a stochastic SSFP problem can be expressed as follows [[19]](#_bookmark30):

*k*

X

max *R*(*X*)= *Ry*(*X*),

*x*∈*S*

*y*=1

Subject to:

*a*11*x*1 + *a*12*x*2 + *a*13*x*3 6 *b*1, *a*31*x*1 + *a*32*x*2 + *a*33*x*3 6 20,

*x*1 + *x*2 + *x*3 6 *b*3,

5*x*1 + 3*x*2 + 4*x*3 6 15, *x*1, *x*2, *x*3 P 0

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

Max *F*(*x*)= *k*1 + *k*2 + *k*3

Subject to:

(*k* + 2*k* + 4*k* — 17)*x* + (*k* + *k* + *k* — 19)*x*

1

2

3

1

1

2

3

2

where *R* (*X*)= *Ny* (*X*)+*ay* , *y* = 1, 2,.. ., *k*.

*y D* (*X*)+*b*

*y y*

+ (*k*1 + 4*k*2 + 7*k*3 — 23)*x*3 + 2*k*1 + 10*k*2 + 5*k*3

+ 1.645q(ﬃﬃ*k*ﬃﬃﬃ2ﬃﬃ+ﬃﬃﬃﬃﬃ0ﬃﬃ.ﬃﬃ5ﬃﬃ*k*ﬃﬃ2ﬃﬃ)ﬃﬃ*x*ﬃﬃﬃ2ﬃﬃ+ﬃﬃﬃﬃﬃ(ﬃﬃ0ﬃﬃ.ﬃ5ﬃﬃﬃ*k*ﬃﬃ2ﬃﬃﬃ+ﬃﬃﬃﬃ2ﬃﬃ*k*ﬃﬃﬃ2ﬃﬃ)ﬃﬃ*x*ﬃﬃ2ﬃﬃﬃ+ﬃﬃﬃﬃ(ﬃﬃ2ﬃﬃ*k*ﬃﬃﬃ2ﬃﬃ+ﬃﬃﬃﬃﬃ3ﬃﬃ*k*ﬃﬃﬃ2ﬃﬃ)ﬃ*x*ﬃﬃﬃ2**ﬃ** 6 12,

*Subject to:*

X*n* !

*p*

*t x* 6 *b*(1)

2 3 1

(4*x*1 + 2*x*2 + 4*x*3)

qﬃﬃﬃﬃﬃﬃﬃﬃﬃ2ﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃ2ﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃ2ﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃ

2 3 2

2 3 3

*ij j i* *i*

P 1 — *p*(1), *i* = 12, ... , *m*; (15)

*j*=1

+ 1.645

0.5*x*1 + 0.25*x*2 + 0.5*x*3 + 0.25 6 12,

qﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃ

X*n* (6*x*1 + 4*x*2 + 6*x*3)+ 1.28

*t x* 6 *b*(2), *i* = *m* + 1, .. . , *h* (16)

1

2

3

*x*1 + *x*2 + *x*3 6 3.16, 5*x*1 + 3*x*2 + 4*x*3 6 15, *x*1, *x*2, *x*3, *k*1, *k*2, *k*3 P 0.

*ij*

*j*

*i*

*j*=1

*x*2 + 0.5*x*2 + 0.75*x*2 6 20,

where 0 6 *Xn*×1 = ¨*xj*¨ ⊂ *Rn* is a feasible set and *R* : *Rn* → *Rn*,

*Test example 3 (SSFP3)*

*Tm*×*n*

¨ *ij* ¨

= *t*(1) ,

2

Max *R*(*X*)=

X

*cy*1*x*1 + *cy*2*x*2 + *ay*

(1)

*b*

*m*×1

= *b*(1) , *i* = 1, 2, ... , *m*, *j* = 1, 2, ... , *n*;

*y*=1 *dy*1*x*1 + *dy*2*x*2 + *by*

Subject to: *a*11*x*1 + *a*12*x*2 + *a*13*x*3 6 27, 5*x*1 + 3*x*2 + *x*3 6 12,

*x*1, *x*2, *x*3 P 0.

*i*

*y*

*y*

*b*(2)

*h*—(*m*+1)×1

= *b*(2) , *i* = *m* + 1, .. . , *h*; *a* , *b*

are scalars.

The deterministic model of the above problem may be given

*Ny*(*X*)= X*cyjxj* and *Dy*(*X*)= X*dyjxj*

¨ *i* ¨

*n*

*n*

*j*=1

*j*=1

Subject to:

¨ ¨

as follows:

Max *F*(*x*)= *k*1 + *k*2

In this model, out of *Ny*(*X*), *Dy*(*X*), *T* and *b*(1) at least one may be a random variable.

*S=*{*X* | Eqs. [(15) and (16)](#_bookmark10), *X* P 0, *X* ⊂ *Rn*} is non-empty,

(20 — 2*k*1 + 4*k*2)*x*1 + (16 — 3*k*1 — 2*k*2)*x*2 + (12 — 5*k*1 — 2*k*2)

— 10*k*1 — 12*k*2

— 1.28q(ﬃﬃ*k*ﬃﬃﬃ2ﬃﬃ+ﬃﬃﬃﬃﬃ*k*ﬃﬃ2ﬃﬃﬃ+ﬃﬃﬃﬃ1ﬃﬃﬃ0ﬃﬃ)ﬃﬃ*x*ﬃﬃﬃ2ﬃﬃ+ﬃﬃﬃﬃﬃ(ﬃ2ﬃﬃﬃ*k*ﬃﬃ2ﬃﬃﬃ+ﬃﬃﬃﬃ*k*ﬃﬃﬃ2ﬃﬃ+ﬃﬃﬃﬃﬃ4ﬃﬃ)ﬃﬃ*x*ﬃﬃﬃ2ﬃﬃ+ﬃﬃﬃﬃﬃ(ﬃﬃ3ﬃﬃ*k*ﬃﬃ2ﬃﬃﬃ+ﬃﬃﬃﬃ2ﬃﬃﬃ*k*ﬃﬃ2ﬃﬃﬃ+ﬃﬃﬃﬃ5ﬃﬃﬃ)ﬃﬃ*x*ﬃﬃ2ﬃﬃ P 3,

1

2

1

1

2

2

1

2

3

convex and compact set in *Rn*.

*Test Example 1 (SSFP1)*

(3*x*1 + 4*x*2 + 8*x*3)+ 1.645q2ﬃﬃﬃ*x*ﬃﬃﬃ2ﬃﬃ+ﬃﬃﬃﬃﬃ*x*ﬃﬃ2ﬃﬃﬃ+ﬃﬃﬃﬃ*x*ﬃﬃﬃ2**ﬃ** 6 27, 5*x*1 + 3*x*2 + *x*3 6 12,

2 *c x* + *c x* + *a*

X *y*1 1 *y*2 2 *y*

Max *R*(*X*)=

*x*1, *x*2, *x*3, *k*1, *k*2 P 0.

1 2 3

*y*=1 *dy*1*x*1 + *dy*2*x*2 + *by*

Subject to: *a*11*x*1 + *a*12*x*2 6 1, *a*21*x*1 + *a*22*x*2 6 *b*2, 16*x*1+

*x*2 6 4, *x*1, *x*2 P 0.

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

Max *F*(*x*)= *k*1 + *k*2

Subject to:

(*k*1 + 2*k*2 — 5)*x*1 + (*k*1 + 3*k*2 — 4)*x*2

qﬃﬃﬃ2ﬃﬃﬃﬃﬃﬃﬃﬃﬃﬃ2ﬃﬃ

*x*1 + *x*2 6 3,

+ 2*k*1 + 4*k*2 + 1.28

For more details on the above examples, please refer [[19]](#_bookmark30).

* 1. *Multi-Objective Stochastic Linear Programming model*

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

The general mathematical model of a constrained MOSLP may be given as [[20]](#_bookmark30):

*n*

X

Maximize *zk* = *ckxj*, *k* = 1, 2, .. . , *k*

*j*

*j*=1

(2*x*1 + *x*2)+ 1.645q*x*ﬃﬃﬃ2ﬃﬃﬃ+ﬃﬃﬃﬃﬃ*x*ﬃﬃ2ﬃﬃ 6 1, (3*x*1 + 4*x*2)

1

2

X*n* X*n* X*n* !

+ 0.84q2ﬃﬃﬃ*x*ﬃﬃﬃ2ﬃﬃ+ﬃﬃﬃﬃﬃ3ﬃﬃﬃ*x*ﬃﬃ2ﬃﬃﬃ+ﬃﬃﬃﬃﬃ2ﬃﬃ 6 3, 16*x*1 + *x*2 6 4, *x*1, *x*2, *k*1, *k*2 P 0.

1

2

*j*=1

Subject to

*P*

*a*1*jxj* 6 *b*1,

*a*2*jxj* 6 *b*2, ... ,

*amjxj* 6 *bm*,

*j*=1

*j*=1

P *P*, *xj* P 0, *j* = 1, 2, .. . , *n*.

*Test example 2 (SSFP2)*

where 0 < *p* < 1 is usually close to 1. It has been assumed that

3 the parameters *a* and *c* are deterministic constants and *b* are

Max *R*(*X*)= X *cy*1*x*1 + *cy*2*x*2 + *ay*

*dy*1*x*1 + *dy*2*x*2 + *by*

*y*=1

*ij j j*

random variables. For more details, the interested reader may

refer to [[20]](#_bookmark30).

*Test example 1(MOSLP1)*

Maximize *z*1 = 5*x*1 + 6*x*2 + 3*x*3, Maximize *z*2 = 6*x*1 + 3*x*2 + 5*x*3, Maximize *z*3 = 2*x*1 + 5*x*2 + 8*x*3

Subject to

*P*(3*x*1 + 2*x*2 + 2*x*3 6 *b*1) P 0.90, *P*(2*x*1 + 8*x*2 + 5*x*3 6 *b*2) P 0.98,

*P*(5*x*1 + 3*x*2 + 2*x*3 6 *b*3) P 0.95, *P*(0.5*x*1 + 0.5*x*2 + 0.25*x*3 6 *b*4) P 0.90,

*P*(8*x*1 + 3*x*2 + 4*x*3 6 *b*5) P 0.99, *x*1, *x*2, *x*3 P 0.

Here, *b*1 follow power Function distribution, *b*2 follow Par- eto distribution, *b*3 follow Beta distribution, *b*4 follow Weibull distribution, and *b*5 follow Bure type XII distribution. The problem is converted to deterministic model as follows:

Maximize *z*1 = *k*1(5*x*1 + 6*x*2 + 3*x*3)+ *k*2(6*x*1 + 3*x*2 + 5*x*3)

+ *k*3(2*x*1 + 5*x*2 + 8*x*3)

Subject to

*Test example 2(MOSLP2)*

Maximize *z*1 = 3*x*1 + 8*x*2 + 5*x*3, Maximize *z*2 = 7*x*1 + 4*x*2 + 3*x*3, Maximize *z*3 = 6*x*1 + 7*x*2 + 10.5*x*3

Subject to

*P*(5*x*1 + 4*x*2 + 2*x*3 6 *b*1) P 0.95, *P*(7*x*1 + 3*x*2 + *x*3 6 *b*2) P 0.95, *P*(2*x*1 + 7*x*2 + 3*x*3 6 *b*3) P 0.95, *P*(2*x*1 + 3*x*2 + 2.5*x*3 6 *b*4) P 0.95, *P*(5*x*1 + 2*x*2 + 1.5*x*3 6 *b*5) P 0.95, *x*1, *x*2, *x*3 P 0.

Here, *b*1 follow power Function distribution, *b*2 follow Par- eto distribution, *b*3 follow Beta distribution of the first kind, *b*4 follow Weibull distribution, and *b*5 follow Bure type XII distri- bution. The problem is converted to deterministic model as follows:

Maximize *z*1 = *k*1(3*x*1 + 8*x*2 + 5*x*3)+ *k*2(7*x*1 + 4*x*2 + 3*x*3)

+ *k*3(6*x*1 + 7*x*2 + 10.5*x*3)

Subject to

*y*2 *y*2 — 100 *y*3 — 5 *e*2*y*4 — 1 3*y*2

5

1

2

5

P 0.95

2

2

9

*y*

10

*e*2*y*4

1 + 3*y*2

1. Experimental settings and numerical results

In this section the computational results of DESP are dis- cussed as well in comparison with other algorithms.

* 1. *Parameter settings*

The three main control parameters of DESP algorithm, popu- lation size *NP* is 50 with exception to MOSLP2 with 75 as the number of decision variables increases (11 decision variables), crossover rate *Cr* is fixed 0.95 and *F* is a random variable fol- lowing uniform distribution as aforementioned. For each algo- rithm, the stopping criterion is to terminate the search process when one of the following conditions is satisfied:

* + 1. the maximum number of generations is reached (assumed 1000 generations),
    2. |*f*max — *f*min| *<* 10—4 where *f* is the value of objective function.

A total of 50 runs for each experimental setting were con- ducted and the best solution throughout the run was recorded as global optimum. Results obtained by the DESP versions are compared with LDE [[2]](#_bookmark17), basic DE, basic PSO and also previ- ously quoted results [[19,20]](#_bookmark30). The results provided by these approaches were directly taken from [[2]](#_bookmark17). Refer to [[2]](#_bookmark17) for more details about LDE algorithms.

* 1. *Numerical results*

DESP algorithm is compared with basic DE, basic PSO and LDE algorithms through various performance metrics like average fitness function value and standard deviation (SD). To compare the convergence speed of algorithms, we consid- ered the average number of function evaluations (NFEs).

* 1. *Result analysis of SSFP*

In order to study the performance of the proposed algorithm DESP on stochastic sum-of-fractional programming (SSFP) problems, performance comparison of DESP algorithm with basic DE, basic PSO, LDE algorithms and results in [[19]](#_bookmark30) is

given in [Tables 1–3](#_bookmark12) in terms of best objective function value

5*x*1 + 4*x*2 + 2*x*3 = *y*1, 7*x*1 + 3*x*2 + *x*3 = *y*2 ,

2*x*1 + 7*x*2 + 3*x*3 = *y*3,

2*x*1 + 3*x*2 + 2.5*x*3 = *y* , 5*x*1 + 2*x*2 + 1.5*x*3 = *y* ,

of deterministic model (*F*(*X*)), optimal decision variable values,

standard deviation and average NFEs.

For the best results obtained by DESP, the constraints are

4

*k*1 + *k*2 + *k*3 = 1,

*x*1, *x*2, *x*3, *y*1, *y*2, *y*3, *y*4 , *y*5, *k*1, *k*2, *k*3 P 0.

5 [0, 0, —0.495616788561048, —0.597382431501665] to ensure

that it is feasible solution, and the objective function value is 1.83246. It can be seen from [Table 1](#_bookmark12) that DESP, LDE1,

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 1 Results of stochastic sum-of-fractional programming problem 1 (SSFP1). | | | | | | |
|  | DE | LDE1 | LDE2 | PSO | Results in [[19]](#_bookmark30) | DESP |
| *F*(*X*) | 1.83245 | 1.83246 | 1.83246 | 1.83218 | 1.7533 | 1.83246 |
| *x*1 | 0.202128 | 0.20199 | 0.202329 | 0.20254 | 0.0602 | 0.202324 |
| *x*2 | 0.165738 | 0.165968 | 0.165426 | 0.164923 | 0.3292 | 0.165433 |
| *k*1 | 1.83245 | 1.883246 | 1.83246 | 1.83218 | 1.7533 | 1.83246 |
| *k*2 | 0 | 0 | 0 | 0 | 0 | 0 |
| SD | 1.2639E—05 | 1.2143E—05 | 1.2147E—05 | 0.120788 | NA | 4.23E—08 |
| Average NFE | 15,925 | 11,790 | 5455 | 37,269 | NA | 4630 |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 2 Results of stochastic sum-of-fractional programming problem 2 (SSFP2). | | | | | | |
|  | DE | LDE1 | LDE2 | PSO | Results in [[19]](#_bookmark30) | DESP |
| *F*(*X*) | 15.2256 | 15.0329 | 15.2256 | 15.2231 | 15.1931 | 15.2256 |
| *x*1 | 0 | 0.0101852 | 0 | 0 | 0 | 0 |
| *x*2 | 1.42539 | 1.76232 | 1.43882 | 1.46877 | 1.4284 | 1.424836 |
| *x*3 | 1.68131 | 1.3875 | 1.67478 | 1.65862 | 1.6818 | 1.681578 |
| *k*1 | 15.2256 | 15.0329 | 15.2256 | 15.2231 | 15.1931 | 15.2256 |
| *k*2 | 0 | 7.278E—08 | 7.268E—08 | 0 | 0 | 0 |
| *k*3 | 6.039E—07 | 5.641E—12 | 0 | 0 | 0 | 0 |
| SD | 1.4346E—05 | 1.2029E—05 | 1.1060E—05 | 1.06941 | NA | 1.03E—05 |
| Average NFE | 38,855 | 4540 | 10,460 | 42,352 | NA | 10,170 |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 3 Results of stochastic sum-of-fractional programming problem 3 (SSFP3). | | | | | | |
|  | DE | LDE1 | LDE2 | PSO | Results in [[19]](#_bookmark30) | DESP |
| *F*(*X*) | 2.32854 | 2.23663 | 2.33083 | 2.23657 | 3.6584 | 2.330846 |
| *x*1 | 1.88002 | 2.39981 | 1.781 | 2.4 | 2.4 | 1.783185 |
| *x*2 | 0.866625 | 0.00031 | 1.03165 | 0 | 0 | 1.028024 |
| *x*3 | 0 | 0 | 0 | 0 | 0 | 0 |
| *k*1 | 2.32854 | 2.23663 | 2.33083 | 2.23657 | 3.6584 | 2.330846 |
| *k*2 | 2.095E—07 | 1.923E—06 | 0 | 0 | 0 | 0 |
| SD | 0.05265 | 0.03543 | 0.03305 | 0.335365 | NA | 0.018375 |
| Average NFE | 5725 | 5350 | 4850 | 44,656 | NA | 5903 |
|  |  |  |  |  |  |  |

LDE2 and DE are able to find the optimal solution consis- tently in all runs but DESP has the lower standard deviation. Moreover, DESP has the better ‘‘best” objective function value than this obtained in [[19]](#_bookmark30) which is slightly infeasible solution. It is to be noted that the improvement percentage of DESP in terms of NFEs in comparison with DE, LDE1, LDE2 and PSO is 70.9%, 60.7%, 15.12% and 87.5%, respectively. There- fore, DESP is considered the most efficient with the smallest (NFE).

For the best results obtained by DESP, the constraints are [0, 0, —1.944514465098703, —0.053585156014804,

—3.999177179444022] to ensure that it is feasible solution, and the objective function value is 15.2256. It can be seen from [Table 2](#_bookmark13) that DESP, LDE2 and DE are able to find the optimal solution consistently in all runs but DESP has the lower stan- dard deviation. Besides, although the improvement percentage of LDE1 in terms of NFEs in comparison with DESP is 55.35%, the best solution obtained by LDE1 is inferior to the best solution obtained by DESP. Moreover, DESP has the better ‘‘best” objective function value than those obtained by PSO and [[19]](#_bookmark30). It is to be noted that the improvement per- centage of DESP in terms of NFEs in comparison with DE, LDE2 and PSO is 73.8%, 0.027% and 75.98%, respectively. Therefore, DESP is considered the most efficient with the smallest (NFE).

For the best results obtained by DESP, the constraints are [0, —13.058526571600890, 0] to ensure that it is feasible solu- tion, and the objective function value is 2.330846. It can be seen from [Table 3](#_bookmark13) that DESP and LDE2 are able to find the optimal solution consistently in all runs but DESP has the lower standard deviation. However, LDE2 is considered the most efficient with the smallest (NFEs). Besides, although the improvement percentage of DE and LDE1 in terms of

NFEs in comparison with DESP is 0.03% and 0.09%, respec- tively, which is not a significant difference for the comparison, the best solutions obtained by DE and LDE1 are inferior to the best solution obtained by DESP. Moreover, DESP has the better ‘‘best” objective function value than that obtained by PSO. The improvement percentage of DESP in terms of NFEs in comparison with PSO is 86.78%. Moreover, it must be noted that the best solution obtained by [[19]](#_bookmark30) is infeasible solution as the constraints are [23.999557387890611,

—14.216684855751021, 0], and the objective function value is 3.6584.

* 1. *Results analysis of MOSLP*

In order to study the performance of the proposed algorithm DESP on Multi-objective Stochastic Linear Programming Problems (MOSLP) problems, performance comparison of DESP algorithm with basic DE, basic PSO, LDE algorithms and results in [[20]](#_bookmark30) is given in [Tables 4 and 5](#_bookmark14) in terms of best objective function value of deterministic model (*F*(*X*)), optimal decision variable values, standard deviation and average NFEs.

For the best results obtained by DESP, the constraints are [—3.077116474765064, —0.000000085756271, —1.4790101

31249243, —0.533838884207691, —3.567129578475143,

—0.000000137249561] to ensure that it is feasible solution, and the objective function value is 12.9312. It can be seen from [Table 4](#_bookmark14) that DESP and PSO are able to find the optimal solu- tion consistently in all runs but DESP has the lower standard deviation. Thus, DESP is more robust than PSO. Besides, the improvement percentage of DESP in terms of NFEs in com- parison with PSO is 43.66%. Additionally, the best solutions obtained by DE and LDE1, LDE2 and the quoted result

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 4 Results of Multi-Objective Stochastic Linear Programming Problem 1 (MOSLP 1). | | | | | | |
|  | DE | LDE1 | LDE2 | PSO | Results in [[20]](#_bookmark30) | DESP |
| *z* | 9.48978 | 10.1553 | 12.0647 | 12.9299 | 8.5089 | 12.9312 |
| *z*1 | 6.18688 | 6.22744 | 6.12031 | 4.84872 | 6.4834 | 4.84872 |
| *z*2 | 9.48978 | 9.34841 | 9.39769 | 8.0812 | 8.3125 | 8.36666 |
| *z*3 | 12.5073 | 12.2764 | 12.4212 | 12.9299 | 10.5140 | 12.9299 |
| *x*1 | 0.352147 | 0.35354 | 0.344071 | 0 | 0.3727 | 2.43E—06 |
| *x*2 | 2.124E—07 | 0.0293907 | 0 | 0 | 0.2319 | 1.48E—06 |
| *x*3 | 1.47538 | 1.4278 | 1.46665 | 1.61624 | 1.0761 | 1.616237 |
| SD | 2.06789 | 1.62183 | 1.95275 | 3.91715 | NA | 1.74E—04 |
| Average NFE | 14,691 | 6932 | 7250 | 20,573 | NA | 11,590 |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table 5 Results of Multi-Objective Stochastic Linear Programming Problem 2 (MOSLP 2). | | | | | | | |
|  | DE | LDE1 | LDE2 | PSO | Results in [[20]](#_bookmark30) | DESP | DESP1 |
| *z* | 6.87235 | 7.13425 | 7.73912 | 7.13425 | 3.2081 | 16.689347 | 15.299629 |
| *z*1 | 3.32379 | 4.18588 | 4.46621 | 4.18588 | 3.8139 | 7.946516 | 7.3175884 |
| *z*2 | 1.99418 | 2.46762 | 2.57927 | 2.46762 | 3.0717 | 4.767910 | 5.5454252 |
| *z*3 | 6.9787 | 8.19198 | 8.12872 | 8.19198 | 5.1968 | 16.687682 | 15.300104 |
| *x*1 | 2.65E—05 | 9.44E—04 | 3.08E—04 | 9.44E—04 | 0.1939 | 9.06E—09 | 0.222094 |
| *x*2 | 1.27E—04 | 0.061029 | 0.127573 | 0.061029 | 0.2810 | 1.23E—07 | 2.08E—05 |
| *x*3 | 0.664552 | 0.738963 | 0.688939 | 0.738963 | 0.1968 | 1.589303 | 1.330228 |
| *y*1 | 1.32963 | 1.72678 | 1.88791 | 1.72678 | 2.4872 | 3.178698 | 3.771069 |
| *y*2 | 0.664947 | 0.928675 | 1.07383 | 0.928675 | 2.3971 | 1.589204 | 2.884872 |
| *y*3 | 1.99454 | 2.64598 | 2.96046 | 2.64598 | 2.9454 | 4.767809 | 4.435026 |
| *y*4 | 1.66177 | 2.03239 | 2.10569 | 2.03239 | 1.7229 | 3.9733268 | 3.769867 |
| *y*5 | 0.9971 | 1.0 | 1.0 | 1.0 | 1.8267 | 2.3839530 | 3.105837 |
| SD | 2.20171 | 1.77604 | 1.41855 | 0.490581 | NA | 4.954E—02 | 3.250082 |
| Average NFE | 30,794 | 50,050 | 50,050 | 50,050 | NA | 64,929 | 61,200 |
|  |  |  |  |  |  |  |  |

[[20]](#_bookmark30), where the problem is solved by Genetic Algorithm (GA), are inferior to the best solution obtained by DESP regardless of their average NFEs. The improvement percentage of DESP in terms of objective function value in comparison with DE, LDE 1, LDE 2 and quoted result [[20]](#_bookmark30) is 136.2%, 127%,

1.07% and 152%, respectively.

Actually, this problems is very difficult to solve as it has the following common features: moderate dimensionality (eleven decision variables) which is almost twice the number of deci- sion variables in MOSLP 1 problem (six decision variables), nonlinear objective functions, more than one equality con- straint (six constraints) and one nonlinear inequality con- straint, and feasibility metric of zero as the feasible region is reduced to a point i.e. the intersection point of all constraints. Thus, it is very difficult to generate feasible solutions during the initial search process with these types of problem. How- ever, by using large tolerance value i.e. by widening the true feasible area, more feasible solutions can be generated to sat- isfy the equality constraints in the initial generation. Then,

43902, —0.000030808461554, —0.000098702360594, —0.000

000037835539] to ensure that it is the feasible and optimal solution (the intersection point of all equality constraints), and the objective function value is 16.689347. Thus, regardless of the average (NFEs) of all compared algorithms, DESP algo- rithm is superior to others in terms of objective function value and it is the only algorithm that is able to find new and the optimal solution to MOSLP 2 problem while the results obtained by others are infeasible solutions. In order to verify the efficiency of the proposed dynamic tolerance rule to handle equality constraints, another version of DESP algorithm, named DESP1, solves the MOSLP 2 problem without using

16

14

**Objective function value**

12

10

through generation, while the initial feasible region is con- tracted, the algorithm improves the previous feasible solutions to satisfy the re-defined feasible space. As a result, it can be deduced that the proposed equality constrained approach is mainly responsible for the poor or good performance of DESP with equality constrained problems with initial tolerance value of *e*(*t*)= 4. From [Table 5](#_bookmark14), it can be obviously seen that the best results obtained by DESP are the optimal solution for this

8

6

4

2

0

DE LDE1 LDE2 PSO Ref.[19] DESP

**Algorithms**

 SSFP1

 SSFP2  SSFP3

challenge problem as the constraints are —0.000000065533720,

—0.000008358091382, —0.000001611376663, —0.0000002908

Figure 3 Performance of DE, LDE1, LDE2, PSO, Ref. [[19]](#_bookmark30) and DESP algorithms in terms of objective function value for SSFPs.

18

16

**Objective function value**

14

12

10

8

6

4

2

0

DE LDE1 LDE2 PSO Ref.[20] DESP

**Algorithms**

 MOSLP1  MOSLP2

rithm performs better in constrained SP problems with different types, complexity and dimensionality; it performs better with regard to the search process efficiency, the final solution quality, the convergence rate, and robustness, when compared with other algorithms. Finally, the performance of the DESP algorithm is superior to and competitive with other compared algorithm. Besides, it is noteworthy to mentioning that DESP algorithm finds the optimal solution for the most difficult MOSLP problem due to the proposed procedure for handling equality constraints. This proves that The DESP algorithm is considered as perfect alternative for solving SP problems. Several current and future works can be developed from this study. Firstly, current research effort focuses on

Figure 4 Performance of DE, LDE1, LDE2, PSO, Ref. [[19]](#_bookmark30) and DESP algorithms in terms of objective function value for MOSLPs.

the proposed rule, applying a constant tolerance value of 0.0001 along the whole search process. [Table 5](#_bookmark14) presents the results of DESP without new dynamic tolerance rule using the same parameter settings. The presented results in [Table 5](#_bookmark14) show that DESP1 was only able to find the best solution which is very close to the optimal solution but it was not able to reach it consistently in all runs as the standard deviation is 3.250082. Consequently, it can be deduced that the proposed dynamic tolerance rule has the main role in solving constrained prob- lems with equality constraints. However, the improvement per- centage of DESP1 in terms of objective function value in comparison with DE, LDE 1, LDE 2, PSO and quoted result

[[20]](#_bookmark30) is 222.62%, 214.45%, 197.69%, 214.45% and 476.9%,

respectively. Thus, it is clearly that DESP1 is superior with others in solving this challenge problem. Based on the above analysis, results and comparisons, it can be concluded that DESP is able to consistently find the global optimal in all stochastic programming problems with a very small standard deviation and with a relatively small average (NFEs) which indicates that the proposed DESP has a remarkable ability to solve considered stochastic programming problems with a perfect performance in terms of high quality solution, rapid convergence speed, efficiency and robustness. Besides, its per- formance is superior and competitive with all compared algo- rithms. Finally, it is noteworthy that the optimal solution of MOSLP 2 which is the most difficulty constrained problem is found. [Figs. 3 and 4](#_bookmark15) show the performance of DESP, DE, LDE, PSO algorithms and the quoted results in [[19,20]](#_bookmark30) in terms of objective function value for SSFPs and MOSLPs, respectively.

1. Conclusions and future work

Stochastic programming (SP) is a framework for modeling optimization problems that involve uncertainty. In this research, two models of SP problems were considered: (i) SSFPs and (ii) MOSLPs problems. The deterministic equiva- lent models of these two classes of stochastic programming models are solved using a new approach to Differential Evolu- tion algorithm, called DESP. The proposed DESP algorithm has been compared with basic DE, basic PSO, LDE1, LDE2 and the quoted algorithm from the literature. The experimen- tal results and comparisons have shown that the DESP algo-

how to control the crossover rate by self-adaptive mechanism. Besides, another benchmark constrained and mixed integer programming problems will be solved using DESP. Addition- ally, future research will investigate the performance of the DESP algorithm in solving unconstrained and constrained multi-objective optimization problems as well as real world applications. Additionally, the promising research direction is joining the proposed triangular mutation with evolutionary algorithms, such as genetic algorithms, harmony search and particle swarm optimization, as well as foraging algorithms such as artificial bee colony, bees algorithm and ant colony optimization.

References

1. [Rao SS. Engineering optimization: theory and practice. 4th ed.](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0005) [Hoboken (NJ): John Wiley & Sons Inc.; 2009](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0005).
2. [Thangaraj R, Pant M, Bouvry P, Abraham A. Solving stochastic](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0010) [programming problems using modified differential evolution](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0010) [algorithms. Log J IGPL 2012;20(4):732–46](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0010).
3. [Carino DR, Kent T, Meyers DH, Stacy C, Sylvanus M, Turner](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0015) [AL, et al. The Russell-Yasuda Kasai model: an asset liability](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0015) [model for a Japanese insurance company using multistage](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0015) [stochastic programming. Interfaces 1994;24(1):29–49](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0015).
4. [Eppen GD, Martin RK, Schrage L. A scenario approach to](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0020) [capacity planning. Oper Res 1989;37(4):517–27](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0020).
5. [Murphy FH, Sen S, Soyster AL. Electric utility capacity](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0025) [expansion planning with uncertain load forecasts. IIE Trans](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0025) [1982;14(1):52–9](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0025).
6. [Fisher M, Hammond J, Obermeyer W, Raman A. Configuring a](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0030) [supply chain to reduce the cost of demand uncertainty. Prod Oper](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0030) [Manage 1997;6(3):211–25](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0030).
7. [Abdelaziz FB, Mejri S. Application of goal programming in a](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0035) [multi-objective reservoir operation model in Tunisia. Eur J Oper](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0035) [Res 2001;133(2):352–61](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0035).
8. [Abdelaziz FB, Aouni B, Fayedh RE. Multi-objective](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0040) [stochastic programming for portfolio selection. Eur J Oper Res](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0040) [2007;177(3):1811–23](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0040).
9. [Mukherjee K, Bera A. Application of goal programming in](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0045) [project selection decision - a case study from the Indian coal](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0045) [mining industry. Eur J Oper Res 1995;82(1):18–25](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0045).
10. [Yang L, Feng Y. A bicriteria solid transportation problem with](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0050) [fixed charge under stochastic environment. Appl Math Mod](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0050) [2007;31(12):2668–83](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0050).
11. [Sen S, Doverspike RD, Cosares S. Network planning with](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0055) [random demand. Telecommun Syst 1994;3(1):11–30](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0055).
12. [Birge JR, Rosa CH. Modeling investment uncertainty in](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0060) [the costs of global CO2](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0060) [Emission policy. Eur J of Oper Res](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0060) [1995;83(3):466–88](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0060).
13. [Abdelaziz FB, Masmoudi M. A multiobjective stochastic program](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0065) [for hospital bed planning. J Oper Res Soc 2012;63(4):530–8](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0065).
14. [Bhattacharya UK. A chance constraints goal programming](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0070) [model for the advertising planning problem. Eur J Oper Res](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0070) [2009;192(2):382–95](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0070).
15. [Charles V, Yadavalli VSS, Rao MCL, Reddy PRS. Stochastic](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0075) [fractional programming approach to a mean and variance model](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0075) [of a transportation problem. Math Prob Eng 2011, 12pages](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0075) [657608](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0075).
16. [Charles V, Dutta D. A method for solving linear stochastic](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0080) [fractional programming problem with mixed constraints. Acta](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0080) [Cienci Indica Math 2004;30(3):497–506](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0080).
17. [Charles V, Dutta D. Linear stochastic fractional programming](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0085) [problem with branch-and-bound technique. In: Proceedings of](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0085) [national conference on mathematical and computational](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0085) [models. Coimbatore (India): PSG College of Technology; 2001.](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0085)

[p. 31–139](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0085).

1. [Charles V, Dutta D, Appal RK. Linear stochastic fractional](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0090) [programming problem. In: Proceedings of international confer-](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0090) [ence on mathematical modeling. India: University of Roorkee;](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0090) [2001. p. 211–7](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0090).
2. Charles V, Dutta D. Linear stochastic fractional programming with sum-of-probabilistic-fractional objective. Optim Online; 2005. [http://www.optimizationonline.org](http://www.optimizationonline.org/).
3. [Charles V, Ansari SI, Khalid MM. Multi-objective stochastic](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0100) [linear programming with general form of distributions. Int J Oper](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0100) [Res Optim 2011;2(2):261–78](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0100).
4. [Suwarna H, Biswal MP, Sinha SB. Fuzzy programming approach](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0105) [to multiobjective stochastic linear programming problems. Fuzzy](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0105) [Set Syst 1997;88(1):173–81](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0105).
5. [Baba N, Morimoto A. Stochastic approximations methods for](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0110) [solving the stochastic multiobjective programming problem. Int J](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0110) [Syst Sci 1993;24(4):789–96](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0110).
6. [Caballero R, Cerda´E, Munoz MM, Rey L, Stancu MIM.](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0115) [Efficient solution concepts and their relations in stochastic multi-](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0115) [objective programming. J Optim Theory Appl 2001;110(1):53–74](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0115).
7. Storn R, Price K. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report TR-95-012. ICSI; 1995.
8. [Storn R, Price K. Differential evolution - a simple and efficient](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0125) [heuristic for global optimization over continuous spaces. J Global](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0125) [Optim 1997;11(4):341–59](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0125).
9. [Engelbrecht AP. Fundamentals of computational swarm intelli-](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0130) [gence. John Wiley & Sons Ltd; 2005](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0130).
10. [Mohamed AW. An improved differential evolution algorithm](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0135) [with triangular mutation for global numerical optimization.](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0135) [Comput Ind Eng 2015;85:359–75](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0135).
11. [Mohamed AW, Sabry HZ. Constrained optimization based on](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0140) [modified differential evolution algorithm. Inf Sci](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0140) [2012;194:171–208](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0140).
12. [Deb K. An efficient constraint handling method for genetic](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0145) [algorithms. Comput Method Appl Mech 2000;186(2–4):311–38](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0145).
13. [Venkatraman S, Yen GG. A generic framework for constrained](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0150) [optimization using genetic algorithms. IEEE Trans Evol Comput](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0150) [2005;9(4):424–35](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0150).
14. [Storn R, Price K. Differential evolution a simple and efficient](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0155) [heuristic for global optimization over continuous spaces. J Global](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0155) [Optim 1997;11(4):341–59](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0155).
15. [Mohamed AW, Sabry HZ, Abd-Elaziz T. Real parameter](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0160) [optimization by an effective differential evolution algorithm.](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0160) [Egypt Inf J (Elsevier) 2013;14:37–53](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0160).
16. [Das S, Suganthan PN. Differential evolution: a survey of the](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0165) [state-of-the-art. IEEE Trans Evol Comput 2011;15(1):4–31](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0165).
17. [Price KV, Storn RM, Lampinen JA. Differential evolution - a](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0170) [practical approach to global optimization. Berlin: Springer; 2005](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0170).
18. Mohamed AW. An efficient modified differential evolution algorithm for solving constrained non-linear integer and mixed- integer global optimization problems. Int J Mach Learn Cyber 2015:1–19. <http://dx.doi.org/10.1007/s13042-015-0479-6>.
19. [Feoktistov V. Differential evolution. In: Search of solutions.](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0180) [Springer; 2006](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0180).
20. [Mohamed AW, Sabry HZ, Khorshid M. An alternative](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0185) [differential evolution algorithm for global optimization. J Adv](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0185) [Res 2012;3(2):149–65](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0185).
21. [Mohamed AW. RDEL: restart differential evolution algorithm](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0190) [with local search for global numerical optimization. Egypt Infom](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0190) [J 2014;15:175–88](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0190).
22. [Brest J. Constrained real-parameter optimization with e-self-](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0195) [adaptive differential evolution. In: Mezura-Montes Efre´n, editor.](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0195) [Constraint-handling in evolutionary computation. Studies in](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0195) [computational intelligence, vol. 198. Berlin: Springer; 2009. p.](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0195) [73–93, chapter 4, ISBN 978-3-642-00618-0](http://refhub.elsevier.com/S1110-8665(16)30036-6/h0195).
23. Charles V. E-model for transportation problem of linear stochas- tic fractional programming. Optim Online; 2007. [http://www.](http://www.optimizationonline.org/) [optimizationonline.org](http://www.optimizationonline.org/).