***I.J. Education and Management Engineering* 2011, 3, 59-66**
Published Online September 2011 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2011.03.09

Available online at http://www.mecs-press.net/ijeme

Binary Encoding Differential Evolution for Combinatorial Optimization Problems

Changshou Deng[a], Bingyan Zhao,Yanlin Yang, Hai Zhang

School of Information Science and Technology, Jiujiang University, Jiujiang City, Jiangxi Province, China

Abstract

Differential Evolution algorithm is a new competitive heuristic optimization algorithm in the continuous field. The operators in the original Differential Evolution are simple; however, these operators make it impossible to use the Differential Evolution in the binary space directly. Based on the analysis of problems led by the mutation operator of the original Differential Evolution in the binary space, a new mutation operator was proposed to enable this optimization technique can be used in binary space. The new mutation operator, which is called semi-probability mutation operator, is a combination of the original mutation operator and a new probability-based defined operator. Initial experimental results of two different combinatorial optimization problems show its effectiveness and validity.

Index Terms: combinatorial optimization problem; Differential Evolution; Binary encoding ; semi-probability mutation operator

1. Introduction

Differential Evolution (DE) is a competitive optimization technique for numerical optimization problems with real parameter [1]. It is a simple yet powerful algorithm for global optimization over continuous spaces, which use the greedy selection criterion to determine which of the rivals to remain in the next generation. Since its invention, the DE algorithm has become quite popular in the machine intelligence and cybernetics. It has been successfully been applied to diverse fields of science and engineering, such as mechanical engineering design [2], signal processing [3] and pattern recognition [4]. It has been proved to perform better than the Genetic Algorithm (GA) or the Particle Swarm Optimization (PSO) by numerical benchmarks experiments [5]. Despite the simplicity and successful application in many engineering fields, its application on the solution of binary optimization problems with binary decision variables is still unusual. One of the possible reasons for this lack is that DE cannot keep the closure when the original DE operators are used in discrete domain directly, for the operators designed in the original DE are designed only for continuous domain. A few works exploit its usage

* Corresponding author:
E-mail address: [a]csdeng@jju.edu.cn

for discrete optimization problems, particularly the combinatorial optimization problem. The Differential Evolution with binary encoding in [6] may be the first version of binary Differential Evolution. In our recent work [7], a novel binary Differential Evolution without scale factor F was proposed.

In this work, the mutation result of the original Differential Evolution was analyzed in depth and then a new semi-probability mutation operator was derived for the binary variables.

The remainder of this paper is structured as follows. Section 2 gives a brief introduction of original DE. A new binary encoding DE (BeDE) is presented in section 3. Two different combinatorial optimization problems used to evaluate this binary encoding DE in section 4. Section 5 concludes this paper.

2. Differential evolution

DE is a population-based stochastic optimizer that starts to explore the search space by sampling at multiple, randomly chosen initial points.

It is a kind of float point encoding evolutionary optimization algorithm. AT present, there have been several variants of DE [1]. One of the most promising schemes, DE/RAND/1/BIN scheme of Storn & Price, is presented in great detail. The pseudo code of DE is given as follows.

```
Initial Population Generation
REPEAT
Mutation
Crossover
Selection
UNTIL (termination criteria are met)
```

In order to clarify the notation used in this paper, the minimization of the objective function $f(x)$ is referred.

2.1 Generation of Initial Population

The DE Algorithm starts with the initial target population $X=(x_{ij})_{m\times n}$ with the size m and the dimension n, which is generated by the following way.

$$x_{ij}(0)=x_j^l+rand(0,1)(x_j^u-x_j^l) \tag{1}$$

where $i=1,2,\cdots,m$, $j=1,2,\cdots,n$, x_j^u denotes the upper constraints, and x_j^l denotes the lower constraints.

2.2 Mutation Operator

For each target vector $x_i(i=1,2,\cdots,m)$, a mutant vector is produced by

$$h_i(t+1)=x_{r1}+F(x_{r2}-x_{r3}) \tag{2}$$

where $i,r_1,r_2\in\{1,2,\ldots,m\}$ are randomly chosen and must be different from each other. And F is the scaling factor which has an effect on the difference between the individual x_{r1} and x_{r2}.

2.3 Crossover Operator

DE employs the crossover operation to add the diversity of the population. The approach is given below.

$$u_i(t+1) = \begin{cases} h_i(t+1), rand \le CR \, or \; j = rand(i) \\ x_i(t), \quad otherwise \end{cases} \tag{3}$$

where $i = 1,2,\cdots,m$, $j = 1,2,\cdots,n$, $CR \in [0,1]$ is crossover constant and $rand(i) \in (1,2,\cdots n)$ is the randomly selected index. In other words, the trial individual is made up with some components of the mutant individual, or at least one of the parameters randomly selected, and some of other parameters of the target individual.

2.4 *Selection Operator*

To decide whether the trial individual $u_i\ (t+1)$ should be a member of the next generation, it is compared to the corresponding $h_i(t+1)$. The selection operation is based on the survival of the fitness among the trial individual and the corresponding one such that:

$$x_i(t+1) = \begin{cases} u_i(t+1), \quad if \quad f(u_i(t+1)) < f(x_i(t)) \\ x_i(t), otherwise \end{cases} \tag{4}$$

DE can adapt itself during the search process and find the optimum efficiently and effectively. The mutation operator can not be used in the binary space directly, while the crossover operator and selection operator can be used in binary space directly.

3. **Binary encoding differential evolution**

A conclusion can be easily drawn from the mutation operator, denoted by the formula (2), that it can only keep the closure in the field of real numbers. However, the original DE cannot keep the closure when it is applied in discrete domain. Thus it cannot be used in discrete optimization problems directly. A Binary-coding DE with new mutation rules was proposed to expand DE into the binary space.

3.1 *Semi-probability Mutation Operator*

A new binary mutation operator was derived from the table of original DE mutation results. Assumed that the binary encoding is used ant the value of F is 1. All the possible results of the original DE mutation operator in binary space directly are shown in table 1.

It is easy to see that there are eight kinds of different combination of the mutually different x_{r1}, x_{r2} and x_{r3}. And 3/4 of the different kinds of combination of the three individuals can achieve binary number, shown in bold form in table 1. Hence this six kinds of combination are eligible even the individual is in binary encoding. Furthermore, the following rules can be derived.

Table 1 Results of original DE operator on binary space

x_{r1}	x_{r2}	x_{r3}	F	results
0	0	0	1	0
0	0	1	1	-1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	1	2
1	1	1	1	1

(a)If x_{r1} equals to zero and x_{r2} equals to x_{r3} ,or x_{r1} equals to one and x_{r2} equals to zero and x_{r3} equals to one, then the result of mutation equals to zero.

(b)If x_{r1} equals to one and x_{r2} equals to x_{r3} ,or x_{r1} equals to one and the three variables are identical, then the result of mutation equals to one.

The results of the rest two kinds of combination are not zero or one. Some modifications must be done to the original DE mutation operator. By further analyzing the remainder forms of the three individual's combination, probability-based rules can be proposed. A probability denoted by pr can be defined as (6).

$$pr = \frac{1}{1+e^{-h_i^{(t+1)}}} \tag{5}$$

The semi-probability Mutation operator in this paper is defined as fellows.

$$h_i(t+1) = \begin{cases} x_{r1} + F(x_{r2} - x_{r3}), & if\ (h_i(t+1) = 0 \\ & or\ h_i(t+1) = 1) \\ 0, & pr < rand \\ 1, & otherwise \end{cases} \tag{6}$$

3.2 Flowchart of the BeDE

The flowchart of the BeDE is given as Fig. 1.

4. Numerical examples

The 0-1 knapsack problem and one-max problem are the typical combinatorial optimization problems. This section will use three knapsack problems with different size and One-Max problem to initially evaluate the BeDE.

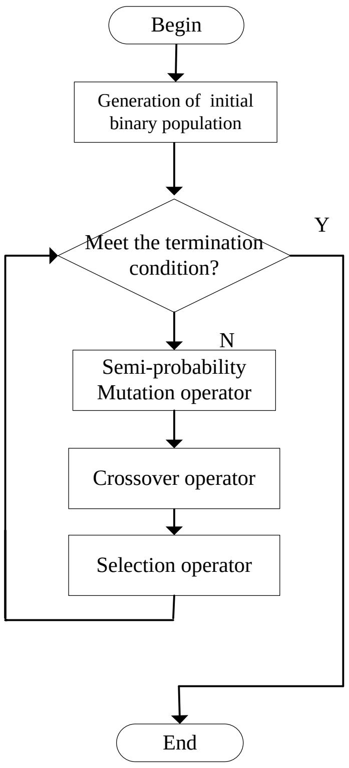
Begin
Generation of initial binary population
Meet the termination condition?
Y
N
Semi-probability Mutation operator
Crossover operator
Selection operator
End

Fig 1. the flowchart of Binary Encoding Differential Evolution

4.1 Knapsack problems

4.1.1 Model of 0-1 knapsack problem

The typical 0-1 knapsack problem (KP) is that there are n given items has to be packed in a knapsack of weight capacity V . Each item has a profit p_i and a weight w_i . The problem is to select a subset of the items whose total profit is a maximized, while whose total weight does not exceed thecapacity V . Then the 0-1 knapsack problem can be formulated as (7) [8].

$$\begin{aligned} &\textit{Maximize} \quad f = \sum_{i=1}^{n} p_i x_i w_i \\ &s.t. \quad \sum_{i=1}^{n} w_i x_i \le V \\ &\qquad x_i \in \{0,1\} \quad (i = 1,2,\cdots n) \end{aligned} \tag{7}$$

4.1.2 Infeasible solution handling

During the evolution of searching the best solution of the knapsack problem, infeasible solutions will appear which means that the conditions in the formula (7) cannot be met. When the total items outrange the capacity of the knapsack, the items with lower profit density will be discarded by priority. In this way, the remainder of the items is closer to the optimum than that of random way. A fixed-up operation is defined to solve this problem. Two steps are adopted in the fixed-up operation. Firstly, all the items are sorted in profit density ascending order. Then one solution is infeasible, the items are discarded in the order of profit density until it is feasible. Numerical experiment

In this section, the BeDE was used to solve the three 0-1 knapsack problems in [7]. The data of the three 0-1 knapsack problems are as fellows.

KP1: n=20

P={92, 4, 43, 83, 84, 68, 92, 82, 6, 44, 32, 18, 56, 83, 25, 96, 70, 48, 14, 58}

W={44, 46, 90, 72, 91, 40, 75, 35, 8, 54, 78, 40, 77, 15, 61, 17, 75, 29, 75, 63}

V=878.

KP2: n=50

P={220, 208, 198, 192, 180, 180, 165, 162, 160, 158, 155, 130, 125, 122, 120, 118, 115, 110, 105, 101, 100, 100, 98, 96, 95, 90, 88, 82, 80, 77, 75, 73, 72, 70, 69, 66, 65, 63, 60, 58, 56, 50, 30, 20, 15, 10, 8, 5, 3, 1}

W={80, 82, 85, 70, 72, 70, 66, 50, 55, 25, 50, 55, 40, 48, 50, 32, 22, 60, 30, 32, 40, 38, 35, 32, 25, 28, 30, 22, 25, 30, 45, 30, 60, 50, 20, 65, 20, 25, 30, 10, 20, 25, 15, 10, 10, 10, 4, 4, 2, 1}

V=1000.

KP3: n=100

P={597, 596, 593, 586, 581, 568, 567, 560, 549, 548, 547, 529, 529, 527, 520, 491, 482, 478, 475, 475, 466, 462, 459, 458, 454, 451, 449, 443, 442, 421, 410, 409, 395, 394, 390, 377, 375, 366, 361, 347, 334, 322, 315, 313, 311, 309, 296, 295, 294, 289, 285, 279, 277, 276, 272, 248, 246, 245, 238, 237, 232, 231, 230, 225, 192, 184, 183, 176, 174, 171, 169, 165, 165, 154, 153, 150, 149, 147, 143, 140, 138, 134, 132, 127, 124, 123, 114, 111, 104, 89, 74, 63, 62, 58, 55, 48, 27, 22, 12, 6}

W={54, 183, 106, 82, 30, 58, 71, 166, 117, 190, 90, 191, 205, 128, 110, 89, 63, 6, 140, 86, 30, 91, 156, 31, 70, 199, 142, 98, 178, 16, 140, 31, 24, 197, 101, 73, 169, 73, 92, 159, 71, 102, 144, 151, 27, 131, 209, 164, 177, 177, 129, 146, 17, 53, 164, 146, 43, 170, 180, 171, 130, 183, 5, 113, 207, 57, 13, 163, 20, 63, 12, 24, 9, 42, 6, 109, 170, 108, 46, 69, 43, 175, 81, 5, 34, 146, 148, 114, 160, 174, 156, 82, 47, 126, 102, 83, 58, 34, 21, 14}

V=6718.

The parameters of the BeDE and the NBDE [7] is presented in table 2.

Table 2 Parameters of Binary encoding DE for the three KPs

Examples	Population size	CR	Maxgen
Kp1	20	0.5	50
Kp2	50	0.5	200
Kp3	50	0.5	1000

For each of the three knapsack problems, 50 trials have been conducted and the best results (Best), average results (Avg), worst results (Worst) and standard deviations (Dev) of the BeDE and NBDE are shown in Table 3.

The results in table 3 show us that BeDE can find the optimums of the three Knapsack Problems with comparable small size of population. The BeDE is better than the NBDE in Avg, Worst, and Dev terms.

Table 3 The statistical results of the three knapsack problems

Problem	Algorith	Bes	Avg	Wors	De
s	m	t		t	v
KP1	BeDE	1042	1041.8	1037	1.8516
	NBDE	1042	1039	1030	3.2482
KP2	BeDE	3119	3116.6	3111	1.2898
	NBDE	3119	3114.4	3102	4.7559
KP3	BeDE	26559	26555.3	26535	5.1912
	NBDE	26559	26550	26529	9.378

4.2 One-Max problem

The aim of a One-Max problem is simply to maximize the ones in a binary string. The fitness of a string is the number of ones it has. The string length 120 is used for this study, with optimum 120.

The parameters set in this study for the One-Max problem is as fellows. Population size is set to 50, $CR = 0.15$ and the maximum generation is 500.

For each of the One-Max problem with length 120, 50 trials have been conducted and the best results (Best), average results (Avg), worst results (Worst) and standard deviations (Dev) of BeDE and NBDE [7] are shown in Table 4.

Table 4 The statistical results of the One-Max problem

Algorithm	Best	Avg	Worst	Dev
BeDE	120	119.96	119	0.1979
NBDE	120	119.84	119	0.3703

The results in talbe IV show us that the proposed BeDE can find the optimum and the performances of Avg , Worst and Dev are slightly better than that of the NBDE.

5. Conclusions

DE is a recently developed algorithm that is very efficient for global optimization over continuous spaces. A binary encoding DE using semi-probability based mutation operator was proposed in this paper to extend the field of DE from the continuous domain to the binary field. The new mutation operator can be applied in binary space directly. Initial experiments on the three different sizes of Knapsack Problems and One-Max problem with size length 120 show that the proposed BeDE is an effective and efficient way to solve combinatorial optimization problems compared with the other one version of binary DE.

Acknowledgment

This work is partially support by the National Natural Science Foundation of China under grant 70771037. It is supported by the Science and Technology Foundation of Jiangxi Province, China under grant No.GJJ10616 as well.

References

[1] Storn, R. and K. Price, "Differential Evolution-A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," Journal Global Optimization,1997,11, pp.241-354.
[2] Rogalsky, Derksen, and Kocabiyik, " Differential evolution in aerodynamic optimization", Proc. of 46th Annual Conf. of Canadian Aeronautics and Space Institute,1999, pp. 29-36
[3] Das S. and Konar A, "Design of tow dimensional IIR filters with modern search heuristics: a comparative study", International Journal of Computational Intelligence and Applications, World Scientific Press, 2006,Vol.6 No.3,pp.176-185.
[4] Das S., Abraham A. and Konar A., "Adaptive clustering using improved differential evolution algorithm", IEEE Transaction on Systems, Man and Cybernetics-Part A, IEEE Press, USA,2008,vol.38, issue 1:pp. 218-237.
[5] Versterstrom J, Thomsen R. " A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithm on numerical benchmark problems "// Evolutionary Computation, CEC2004.Portland OR: IEEE press, 2004,2: 1980-1987
[6] T. Gong, L. T. Andrew. "Differential Evolution for Binary Encoding". Soft Computing in Industrial Applications, ASC 39, pp. 251-262, 2007.
[7] Changshou Deng, Bingyan Zhao, Yanling Yang, etc. Novel binary Differential Evolution without scale factor F, Third International Workshop on Advanced Computational Intelligence, 2010,8: 250-253.
[8] Y. Liu and C. Liu.: "A Schema-Guiding Evolutionary Algorithm for 0-1 Knapsack Problem". Iacsit-sc. In: 2009 International Association of Computer Science and Information Technology –Spring Conference, IEEE Press, Singapore,pp.160-164, 2009.