Modified Binary Differential Evolution for Solving Wind Farm Layout Optimization Problems

Dazhi Jiang, Chenfeng Peng
Department of Computer Science
Shantou University
Shantou, China
{dzjiang, 11cfpeng}@tom.com

Zhun Fan*, Yan Chen
Department of Mechatronics Engineering
Shantou University
Shantou, China
{zfan, ychen}@ stu.edu.cn

Xinye Cai
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics,
xinye@nuaa.edu.cn

***Abstract*—an important phase of a wind farm design is solving the wind farm layout optimization problem, for which evolutionary algorithms have been used frequently. According to the characteristics of wind farm layout problems, a smoothing operator is presented and adopted into the binary differential evolution. The experimental results show that the new algorithm effectively improves solution quality with less execution time. In addition, irregular wind field layout problem is studied and solved in this paper.**

***Keywords—Wind Farm Optimization; Binary Differential Evolution; Smooth Operator; Irregular Wind Field*.**

I. INTRODUCTION

Nowadays, the use of renewable energies is increasing all over the world as important alternatives to generate free and clean power. Wind energy is one of the best choices of renewable energy. Most wind energy is produced by wind farm. A wind farm is a set of wind turbines, whose positions should be decided in order to maximize the produced energy. Wake effect is a key influencing factor to produce energy in wind farm. Due to the wake effect, the downstream turbines have lower wind speeds and less energy capture if they are located behind a wind turbine. However, the wake effects can be reduced by optimizing the geometry of the wind farm, i.e. the locations of all individual wind turbines of the wind farm. Evolutionary algorithms have been used widely to solve this kind of problem. Mosetti et al. [1] were the first to obtain the optimal wind farm position based on genetic algorithm. In [2], Grady et al. replicated the experiments using modified settings of the GA. The result showed that the genetic algorithm can achieve better solutions by having 20 subpopulations evolve for 3000 iterations. Recently, Housheng [3] improved these results using a distributed GA where a small fraction of the highest quality individuals of each subpopulation periodically migrate to other subpopulations. Sisbot et al. [4] proposed a multi-objective genetic algorithm, where two objectives - maximizing the power produced and minimizing the cost are considered simultaneously. Particle swarm optimizer (PSO) is an effective optimization technique, which was firstly presented by Kennedy and Eberhat in 1995[5]. As a novel and effective algorithm, PSO has also been successfully applied in wind farm optimization. Chunqiu Wan et al. [6] demonstrated that the PSO approach is more suitable and effective for micro-siting than the classical binary coded genetic algorithms. Martin Bilbao and Enrique Alba [7] adopted the GPSO (Geometric Particle Swarm Optimization) and showed in experiments that GPSO achieved the same sitting in less execution time and less number of evaluations. Similar to the traditional evolutionary algorithm (EA), Differential Evolution (DE) [8] is a reliable and effective global optimizer algorithm which has been investigated and applied extensively. However there are few papers adopting DE algorithm or binary DE to solve layout optimization problems. In this paper, we presented a modified binary differential evolution algorithm for solving layout optimization problems, named as binary differential evolution based on smoothing operator (BDESO), where smoothing operator was designed according to the characteristic of wake model. The experimental results show that the new algorithm effectively improves solution quality in less execution time.

*corresponding author

Organization of this paper is as follows. In Section II, some models with regards to wind farm layout problems are introduced. In Section III, the Modified Binary Differential Evolution is presented and explained. Experimental verifications are provided in Section IV. The last Section gives conclusions and discusses future research directions.

II. WIND FARM MODELS

In this section, four models are adopted as components to deal with the wind farm layout problem, which are encoding model, wake effect model, power output model, and cost model. The detailed information of each model is described as follows.

A. Encoding Model

Binary coding evolutionary algorithms are frequently adopted to solve wind farm layout problem. In binary coding evolutionary algorithm, each individual of the population is a binary vector $V_i=\{V_{i1},V_{i2},\cdots,V_{in}\}$ representing the terrain (Fig.1) where the wind farm will be installed; each element V_{ij} can has a wind turbine (represented by 1) or be empty (represented by 0). For example in Fig.1, the first row can be coded as 1010011010, and each individual has a length of 100 elements (10*10 grids). The length of vector (solution space) is decided by the shape of terrain, and can be changed by the user.

B. Wake Effect Model

We adopt Jensen model [1] as wake effect model for simplification of the wind field calculations. This model assumes the wake radius r_1 of the wake increases linearly, proportional to the downstream distance x as shown in Fig. 2.

This model is more practical to embed within optimization procedures and in computer programs [9].

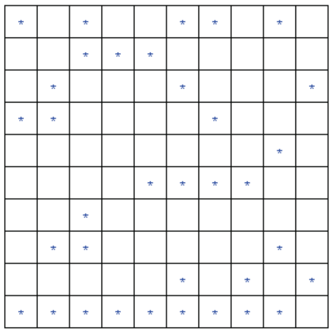

Figure 1. Example of wind farm layout('＊'represents the cell in the wind farm grid contains a turbine)

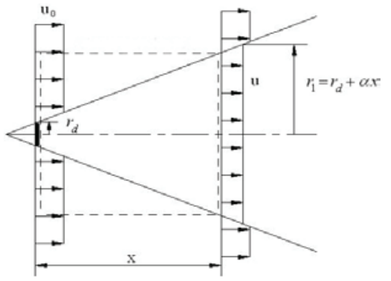

Figure 2. Jensen wake model [1]

After performing a momentum balance and using Betz theory to determine the wind speed immediately behind the rotor [2], the following expression is derived to describe the wind speed downstream of the turbine.

$$U_j = U_0 \left(1\text{-}vd_{ij}\right) \tag{1}$$

where U_0 is the initial wind speed, vd_{ij} is the *velocity deficit* induced on position j by the wake generated by i. vd_{ij} is computed as follows:

$$vd_{ij} = \frac{2\alpha}{(1+\alpha(x_{ij}/r_d))^2} \tag{2}$$

where x_{ij} is the distance between position i and position j. The parameter α is called *wake expands constant* and is computed by following expression:

$$\alpha = \frac{1}{2\ln(z/z_0)} \tag{3}$$

where z is the hub height of the turbine and z_0 is a constant called *surface roughness*, which depends on the terrain characteristics. α determines how quickly the wake expands with distance.

The term r_d is called *downstream rotor radius* which equals to:

$$r_d = r\sqrt{\frac{1\text{-}a}{1\text{-}2a}} \tag{4}$$

The parameter a in the expression is called *axial induction factor* and is derived by the following expression:

$$a = 0.5\left(1\text{-}\sqrt{1\text{-}C_T}\right) \tag{5}$$

The parameter C_T is called *thrust coefficient.*

In a wind farm where many turbines are installed, one turbine is likely affected by several others upwind at the same time. In the Jensen model, the total velocity deficit $v_{def}(j)$ at location j that is affected by more than one wakes is obtained as follows:

$$v_{def}(j) = \sqrt{\sum_{i\in W(j)} vd_{ij}^{\ 2}} \tag{6}$$

where $W(j)$ is the set of turbines affecting position j with a wake. $v_{def}(j)$ is used in Eq.(2) in place of vd_{ij} to compute U_j. Fig.3 is a sample illustration about multiple wakes.

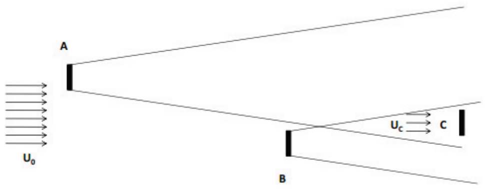

Figure 3. Example of multiple wakes

The overall calculation process is as follows. First, Eq.(2), (3), (4) and (5) are used to compute vd_{AC} and vd_{BC}. Then Eq.(6) is used to get $v_{def}(j)$. Finally, Eq. (1) is applied to get U_C.

From Eq.(2), we find that *velocity deficit* is inversely proportional to x_{ij}. In other words, the total velocity deficit depends mostly on the closest turbine that generates a wake. The calculation is the same if B is inside the wake created by A in Fig. 3.

C. Power Output Model

In this work, the power output model is described as follows [1]:

$$P_i(U)(KW) = \begin{cases} 0, & U<2m/s \\ 0.3U^3, & 2m/s \le U<12.8m/s \\ 629.1, & 12.8m/s \le U \le 18m/s \\ 0, & U>18m/s \end{cases} \tag{7}$$

$$P_{tot} = \sum_{i=1}^{N} P_i \tag{8}$$

where $P_i(U)$ represent the power of turbine i. U is the wind speed on the wind turbine i. P_{tot} is the total power generation for all wind turbines in the wind farm. N is the total number of wind turbines.

D. Cost Model

Mosetti [1] assumed that in the wind farm, the cost/year of turbines will reduce with the number of turbines increasing, and the maximum reduction will reach 1/3 of the cost of turbines. The cost model can be expressed as follows.

$$cost=N\left(\frac{2}{3}+\frac{1}{3}e^{-0.00174N^2}\right) \tag{9}$$

$$Cost_{tot}=cost_y \times cost \tag{10}$$

where $cost_y$ is the cost of one turbine per year. $Cost_{tot}$ is total cost of turbines in the wind farm per year.

III. Binary DE based on Smooth Operator

The original Differential Evolution (DE) performs well in continuous-valued search spaces. But the manner, in which the operator is defined, makes it practically impossible to be applied to binary spaces without any improvement. The Angle Modulated DE (AMDE) [10] is a kind of modified DE to solve binary optimization problems. The basic idea of AMDE is using a mapping method to enable the DE to operate within binary spaces. In this line of thought, a mapping method is presented in this paper and given as follows.

$$B_{i,j}=round\left(\left(V_{i,j}-x_j^L\right)/\left(x_j^U-x_j^L\right)\right) \tag{11}$$

B_i is the binary string of V_i after mapping, $i\in\{1,2,\cdots,N\}$, $j\in\{1,2,\cdots,D\}$, $\left[x_j^L,x_j^U\right]$ is the Domain of x_j , $round(x)=[x]$.

Assume N is a constant number which presents the size of population, and D is the dimension of parameter vectors. The population is expressed as $V_i(t)$, where $i=1,2,\cdots,N$, t is the generation. For the new algorithm called BDESO, the procedure is illustrated in the following.

Mutation: For each vector *i* in generation *t*, a mutant vector $V_i(t+1)$ is defined by:

$$V_i(t+1)=V_{r_1}(t)+F(V_{best}(t)-V_{r_1}(t))+F(V_{r_2}(t)-V_{r_3}(t)) \tag{12}$$

where $i\in\{1,2,\cdots,N\}$ and r_1 , r_2 , r_3 $\in[0,N]$, i, r_1 , r_2 and r_3 are different. The differential mutation parameter *F*, known as scale factor, is a positive real normally between 0 and 1. Simply, larger values for *F* result in higher diversity in the generated population and the lower values in faster convergence.

Crossover: Crossover plays an important role in DE algorithm which increases the diversity of the population. A crossover vector $V_i'(t+1)$ is defined as following.

$$V_i'(t+1)=\left(V_{i,1}'(t+1),V_{i,2}'(t+1),\cdots,V_{i,D}'(t+1)\right)$$
$$V_{i,j}'(t+1)=\begin{cases}V_{i,j}(t+1), & if\ rand()<P_c\ and\ j{=}k\\ V_{i,j}(t), & else\end{cases} \tag{13}$$

The recombination probability parameter $P_c\in[0,1]$,

$rand()\in[0,1]$, $j\in\{1,2,\cdots,D\}$, $k=rand(0,D)$.

Smoothing: In order to improve the result of optimization, we create the smoothing operator for the algorithm.

$$V_{i,j}=V_{i,j}-\sigma\cdot\left(V_{i,j}-\left(V_{i,j-1}+V_{i,j+1}\right)/2\right) \tag{14}$$

where σ is a smoothing factor, which controls the smoothing strength, $\sigma\in[0,1]$.

The improved BDE algorithm, BDESO is designed with a focus on smoothing strategy. The expression shows that in a vector, every dimension gets close to the average of the two dimensions beside it. So between each dimension there is a smooth transition. The function of smoothing operator could be interpreted according to the actual situation of wind farm layout. In the wind farm layout optimization problem, turbines are non-interfering with each other in the horizontal direction perpendicular to the wind. However, in the vertical direction parallel to the wind, turbines will generate wake effect which may affect downstream turbines.

Selection: The selection strategy of DE is as follows.

$$V_i(t+1)=\begin{cases}V_i'(t+1), & if\ f\left(V_i(t)\right)\geq f\left(V_i'(t+1)\right)\\ V_i(t), & else\end{cases} \tag{15}$$

If the individual $V_i'(t+1)$ is better than $V_i(t)$, then $V_i(t)$ is replaced by $V_i'(t+1)$, otherwise the $V_i(t)$ is retained.

The pseudo code of BDESO Algorithm is provided in Fig. 5.

```
    Algorithm of BDESO
1:  t←0;/*Generation*/
2:  Initializes a population N and set the parameter-values
3:  Every individual generates its Binary String (using Eq.(11)) and
    evaluate its fitness
4:  repeat
5:    for each individual $V_i(t)$ of N do
6:        $V_i(t+1)$ ←mutation( $V_i(t)$ )
7:        for each dimension $V_{i,j}(t)$ do
8:          if(rand()<Pc or j==rand(0,D)) then
9:              $V_{i,j}'(t+1) \leftarrow V_{i,j}(t+1)$
10:         else
11:             $V_{i,j}'(t+1) \leftarrow V_{i,j}(t)$
12:         end if
13:       end for
14:   end for
15:   for each individual $V_i'(t+1)$ do
16:       if (rand()<$P_{si}$) then
```

```
17:            for each dimension $V_{i,j}'(t+1)$ do
18:               if(rand()<$P_{sd}$) then
19:                  $V_i'(t+1)$ ←smooth($V_{i,j}'(t+1)$)
20:               end if
21:            end for
22:         end if
23:      end for
24:      Every offspring individual generates its Binary String (using
         Eq.(11)) and evaluates its fitness
25:      if $f(V_i'(t+1)) < f(V_i(t))$ then
26:         $V_i(t+1) \leftarrow V_i'(t+1)$
27:      else
28:         $V_i(t+1) \leftarrow V_i(t)$
29:      end if
30:   t←t+1
31:   until Stopping condition is met
```

Figure 4. The pseudo code of BDESO Algorithm

IV. EXPERIMENTAL STUDY

A. Uniform and Single Direction Wind Speed of 12m/s

In this section, we assume the uniform wind coming from north and the wind speed is 12m/s. BDESO (binary Differential Evolution without Smoothing operator) and GA will be applied to wind farm layout optimization. For each algorithm, 20 independent runs were conducted with 300,000 function evaluations (FES) as the termination criterion. The result of the experimental will be given, including best solution, average fitness values, total annual power output, number of wind turbines, average efficiency of wind farm and the number of evaluation used to find the best solution. In this paper, the parameter settings are shown in Table 1.

TABLE I. PARAMETERS SETTING IN BDESO, BDE AND GA

Parameter	Description	BDESO	BDE	GA
N	population size	600	600	600
r	rotor radius	20m	20m	20m
z	hub height	60m	60m	60m
z_0	surface roughness	0.3	0.3	0.3
C_t	thrust coefficient	0.88	0.88	0.88
$Cost_y$	The cost per turbine per year	3.2E6￥	3.2E6￥	3.2E6￥
P_c	crossover probability	—	—	0.7
P_m	Mutation probability	0.5	0.5	0.2
F	Mutate factor	0.3	0.3	—
δ	Mutate factor	0.2	0.2	—
P_{si}	Individual smoothing probability	0.2	—	—
P_{sd}	Dimension smoothing probability	0.6	—	—
σ	Smoothing factor	0.6	—	—

In this paper, the objective function is defined as follows [2].

$$fitness = \frac{cost}{P_{tot}} \qquad (16)$$

The objective function will minimize the cost per unit energy produced.

The profit function is defined as follow:

$$profit = P_{tot} \cdot price - Cost_{tot} \qquad (17)$$

where *price* is the price of a KWh of electrical energy on the market in ￥, and its value is 0.8￥.

The best configuration of the farm found for each algorithm is illustrated in Fig. 5. It can be found that the best layout uses 30 wind turbines and they are aligned in rows keeping a constants distance between them, and the rows perpendicular to the wind direction.

Fig. 6 shows the fitness evolution curve of each algorithm. In the early stages of evolution, BDESO and GA have a similar convergence speed. But during the late stages BDESO is better and can find the optimal solution more quickly. However, BDE could not compete with GA, without the smoothing operator.

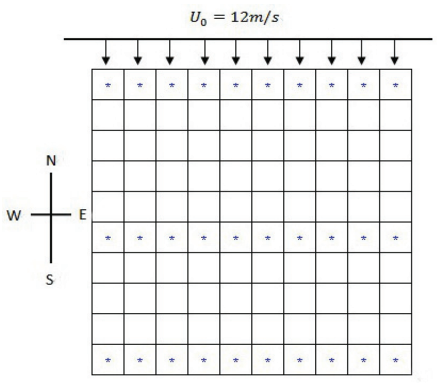

Figure 5. Best configuration of wind farm

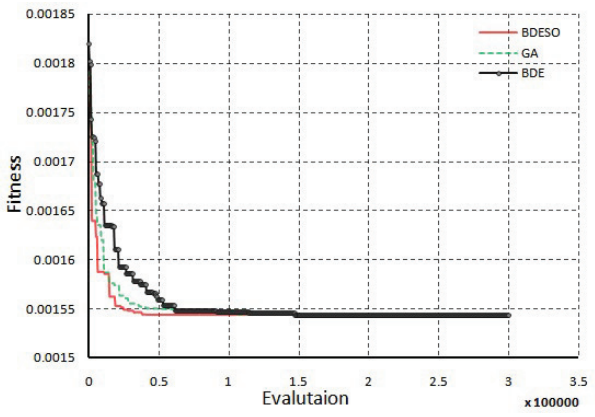

Figure 6. Fitness evolution curve

TABLE II. RESULT OF EXPERIMENTAL

Description	BDESO	GA	BDE
Best Fitness	0.001543341	0.001543341	0.001543341
Average Fitness	0.001543347	0.001543434	0.001544231
Average Profit(￥)	29616219.68	29610541.44	29801289.6
Average Power Output(KW)	125375435.7	1253368338.2	126333748.2
Average Efficiency	92.028%	92.023%	91.673%
Number of Wind Turbines	30	30	30
Average Evaluation of Best Solution Found	89390	195640	344160

From Table Ⅱ we can see that BDESO, BDE and GA obtained the same best fitness value. However, The BDESO obtained better average power output, average profit, average efficiency and the fewer average evaluations to obtain the best solution. The result indicates the BDESO is effective on average solution quality and execution time.

In order to keep the population's diversity, and avoid local optimum, every individual will be selected for smoothing based on a probability P_{si} . Furthermore, every dimension in a selected individual is also smoothed according to a probability P_{sd}.

Fig.7 and Fig.8 shows the fitness evolution curves with different smoothing factors. We found that the convergence speed of our algorithm is increased with the increase of smoothing factor. However, when the smoothing factor is too large, the algorithm is easy to fall into local optimum. According to the experiments, we found that σ=0.6 is proper for BDESO.

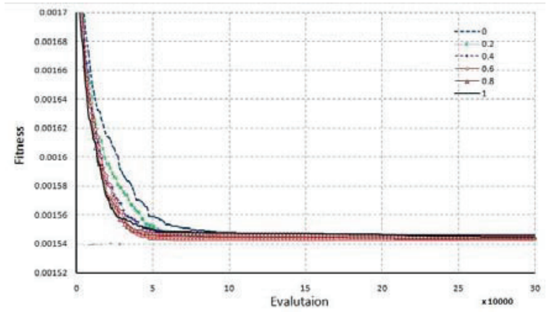

Figure 7. Fitness evolution curves with different smoothing factors

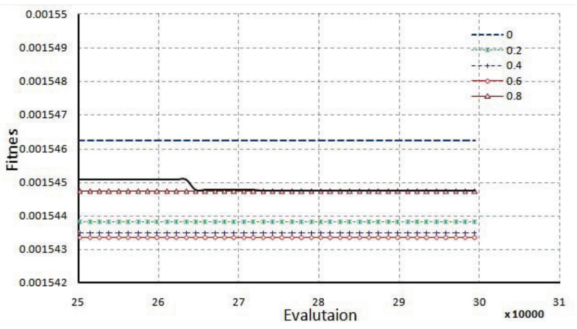

Figure 8. A partial of fitness evolution curves with different smoothing factors

B. *The Layout of Different Shapes of Wind Farms*

In the real world, the shape of most wind fields are not regular rectangle because of the terrain factors such as lakes, rivers, gullies, and some building, which the wind turbine can't be built in. In this paper, we analyze the characteristics of wind farm layout according to different situations of the wind field.

The following is the optimization result of circular wind field and irregular wind field. From Fig. 9 (in Fig.9-12, 'X' represents the cell in the wind farm grid con not contain a turbine) and Fig. 10, we can see that most wind turbines were distributed in the edge area, and only a few were allocated in the center area. That's because the wake effect is stronger in the center area.

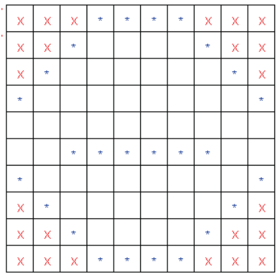

Figure 9. Circular wind field

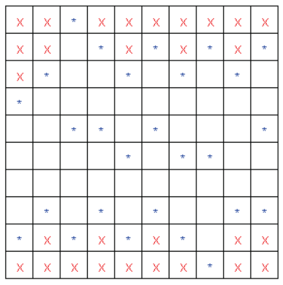

Figure 10. Irregular wind field

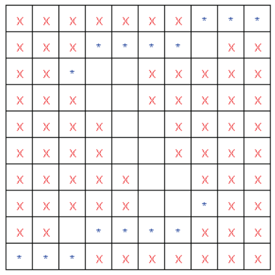

Figure 11. Narrow wind field

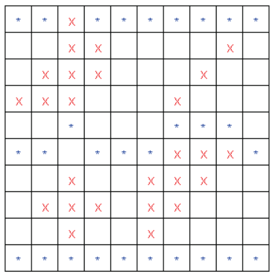

Figure 12. Wind field with blocks

Fig.11 shows a narrow wind field and the turbines are distributed in both ends but center to avoid being affected by the wake effect. It is noted that the blocks in the Fig.12 stand for lakes or lower building, but not mountains or some higher obstacles. According to that, we can easily understand the rationality of the wind farm layout in Fig.12.

V. CONCLUSION AND FUTURE WORK

We have solved the wind farm layout problems with BDESO algorithm, and the experimental result demonstrates that the BDESO is efficient. By comparing BDESO with BDE, we found the smoothing operator is effective in solving wind farm layout optimization problems. It can help the algorithm find the best solution more quickly without indulging into local optimal. Furthermore, the algorithm also gets good result to solve the layout optimization problem of wind farm with blocks or different shapes. The layout problem of complicated terrain will be more meaningful. However, according to the experiments we find that the structure of initial population and the parameters setting affect the efficiency of the presented algorithm, one future work will focus on the parameters study and adjustment in BDESO algorithm.

Wind farm layout optimization is a complicated problem. More real world factors should be considered, such as wind multi-direction, complicated terrain, evaluation of annual average wind resource and so on. These will be the subjects of the future study.

ACKNOWLEDGEMENTS

The authors would like to thank anonymous reviewers for their very detailed and helpful review. This work was supported by the National Natural Science Foundation of China (No.: 61175073) , in part by the Leading Talent Project of Guangdong Province, in part by the Shantou Science and Technology Planning Project (No.:150), and Li Ka-Shing Foundation project.

REFERENCES

[1] Mosetti G, Poloni C, and Diviacco D, "Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm," Journal of Wind Engineering and Industrial Aerodynamics, 51(1), pp.105-116,1994.

[2] Grady SA, Hussaini MY, and Abdullah MM, "Placement of wind turbines using genetic algorithms," Renewable Energy, 30(2), pp.259–270, 2005.

[3] Hou-Sheng H, "Distributed Genetic Algorithm for Optimization of Wind Farm Annual Profits," International Conference on Intelligent Systems Applications to Power Systems, Kaohsiung, Taiwan, pp.418–423, November 2007.

[4] Sisbot S, Turgut O, Tunc¸ M, et al, "Optimal positioning of wind turbines on Gokceada using multi-objective genetic algorithm," Wind Energy, 13(4), pp.297-306, 2010.

[5] Kennedy J, and Eberhart RC, "Particle swarm optimization," Proc. of the IEEE Conf. on Neural NetWork, IV.Perth: IEEE Press, pp.1942-1948, 1995.

[6] Chunqiu Wan, Jun Wang, Geng Yang, and Xing Zhang, "Optimal Micro-siting of Wind Farms by Particle Swarm Optimization," ICSI (1) , pp.198-205, 2010.

[7] Bilbao, M., and Alba, E., "GA and PSO Applied to Wind Energy Optimization," CACIC Conference Proceedings, Jujuy, Argentina,Oct. 2009.

[8] R. Storn, and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," Journal of Global Optimization, 11(4), pp.341-359, Dec 1997.

[9] Samorani, M., "The Wind Farm Layout Optimization Problem," PowerLeeds School of Business. http://bit.ly/fMzqBZ, 2010.

[10] Pampará G., Engelbrecht A.P., and Franken N., "Binary Differential Evolution," IEEE Congress on Evolutionary Computation , pp. 1873-1879 , 2006.