Windmill Farm Pattern Optimization using Evolutionary Algorithms

Charlie Vanaret
ENAC[*], IRIT[†]
7 av Belin
31055 Toulouse Cedex 5, France
vanaret@recherche.enac.fr

Nicolas Durand
ENAC, IRIT
7 av Belin
31055 Toulouse Cedex 5, France
durand@recherche.enac.fr

Jean-Marc Alliot
IRIT
118 route de Narbonne
31062 Toulouse Cedex 9, France
jean-marc.alliot@irit.fr

ABSTRACT

When designing a wind farm layout, we can reduce the number of variables by optimizing a pattern instead of considering the position of each turbine. In this paper we show that, by reducing the problem to only two variables defining a grid, we can gain up to 3% of energy output on simple examples of wind farms dealing with many turbines (up to 1000) while dramatically reducing the computation time.

Categories and Subject Descriptors

J.6 [**Computer-Aided Engineering**]: Computer-aided design; D.2.2 [**Software Engineering**]: Design Tools and Techniques—*Computer-aided software engineering*

Keywords

Wind Energy, Wind Farm Layout, Optimization

1. INTRODUCTION

In the last 15 years, the attempts to discover techniques for efficiently installing wind farms both onshore and offshore have increased considerably; the recent review of Gonzalez [1] lists almost 150 bibliographic references for the optimal wind-turbine micro-siting problem.

In this paper, we use the park model [3] to compute wake effects. The wake effects on a turbine i created by all turbines j ($j \neq i$) change the wind resource available to i by reducing the scale parameter c of the Weibull distribution estimated for the entire farm, called the freestream wind resource. These effects depend on the relative location of the j turbines from i. Thus, there exists a parameter c_i for each turbine i: its computation is complex and involves wind velocity deficits $Vdef_{ij}$ that the turbine i experiences due to the influence of other turbines j. The simple evaluation of one configuration is thus quadratic with respect to the number of turbines. The algorithm that we use to compute the wake effects is described in [4]. Among the best known results so far, Wagner [5] uses CMA-ES to place 1000 turbines, but it takes weeks to solve the problem.

In this paper, we use the exact models and algorithms[1] from [4] and show that using regular patterns optimized by an evolutionary method outperforms current evolutionary and deterministic methods: the computation times are decreased and the power output is improved on large instances (over 200 turbines). We argue that when the number of variables becomes too large, the ES gets stuck in local minima whatever the computation time, thus rendering methods with fewer variables more efficient. To prove our point, we employ here a very simple regular pattern model using only two variables.

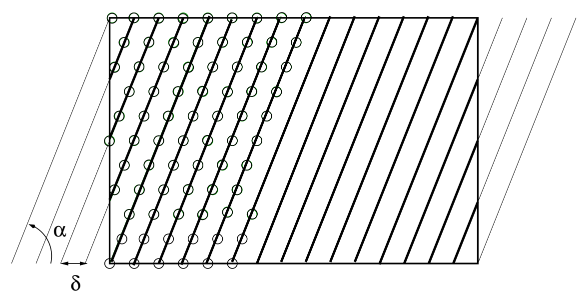

Figure 1: Windmill farm pattern modeling

2. MODELING AND OPTIMIZATION

[6] and [2] have already studied simple instances of small windmill farms (< 100 windmills) placed on a grid.

Here, we define a simple windmill farm pattern with two parameters to optimize the windmills' positions on a field such as that presented in [4] (see figure 1). α determines the angle of the windmills' alignment with the longest edge of the field. δ measures the space between two lines of windmills on the longest edge of the field. We set the n windmills up as regularly as possible. A windmill is positioned at each edge of the field. If n is the total number of windmills, l the total length of the bold segments and n_s the number of bold segments, then the distance between two windmills is close

[*]Ecole Nationale de l'Aviation Civile

[†]Institut de Recherche en Informatique de Toulouse

[1]We want to thank Sylvain Cussat-Blanc for giving us access to the source code of [4, 7].

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-2881-4/14/07.
http://dx.doi.org/10.1145/2598394.2598506

size n	TDA 200k max e+05	TDA 200k mean e+05	time min	GA max e+05	GA mean e+05	time min	α *opt*	δ *opt*	gain +% loss -%	DE max e+05	DE mean e+05	time min	α *opt*	δ *opt*	gain +% loss -%
10	0.7314	0.7309	13	**0.7315**	**0.7314**	6	810.14	0.4952	**0.07**	**0.7315**	0.7313	6	809.93	0.4951	**0.05**
20	1.449	1.448	30	**1.456**	**1.454**	2	789.71	0.5311	**0.41**	**1.456**	**1.454**	3	790.21	0.5309	**0.41**
30	2.144	2.135	45	**2.164**	2.162	2	746.27	0.0381	**1.26**	**2.164**	**2.164**	3	745.82	0.0381	**1.36**
40	2.806	2.791	61	**2.814**	2.808	3	919.46	0.5924	**0.61**	**2.814**	**2.809**	3	918.81	0.5920	**0.64**
50	3.418	3.412	77	**3.461**	**3.461**	3	745.07	0.7370	**1.44**	**3.461**	3.460	3	744.85	0.7369	**1.41**
60	**4.022**	**4.011**	95	3.988	3.954	4	692.07	0.7452	-1.42	3.988	3.984	4	691.87	0.7449	-0.67
70	**4.591**	**4.555**	111	4.449	4.443	5	566.51	0.7264	-2.46	4.449	4.446	4	566.21	0.7261	-2.39
80	**5.108**	**5.090**	129	4.925	4.921	7	499.77	0.6743	-3.32	4.924	4.919	5	499.38	0.6739	-3.36
90	**5.625**	**5.609**	151	5.450	5.419	7	431.46	0.7257	-3.39	5.450	5.441	6	431.32	0.7260	-3.00
100	**6.113**	**6.083**	170	5.863	5.796	6	395.30	0.6456	-4.72	5.862	5.819	6	395.03	0.6443	-4.34
200	13.25	13.23	404	**13.59**	**13.59**	16	727.00	0.6883	**2.72**	**13.59**	**13.59**	14	727.05	0.6882	**2.72**
300	19.73	19.71	678	**20.25**	20.24e	26	735.16	0.7411	**2.69**	**20.25**	**20.25**	23	735.13	0.7411	**2.74**
400	25.86	25.84	1020	**26.66**	**26.65**	39	722.81	0.6934	**3.13**	**26.66**	26.64	35	722.76	0.6932	**3.10**
500	32.51	32.49	1470	**33.52**	**33.52**	53	724.88	0.7399	**3.17**	**33.52**	33.50	50	724.75	0.7399	**3.11**
10^3	64.54	64.49	4500	**66.69**	**66.68**	166	709.31	0.6940	**3.40**	**66.69**	**66.68**	162	709.26	0.6940	**3.40**

Table 1: Comparison of TDA, GA and DE for different numbers of turbines (time is for 30 runs, as in [4])

to $d = \frac{l}{n-n_s}$. For fixed δ and α, a simple process determines the positions of the windmills regularly on the field.

We chose evolutionary algorithms (EA) for solving this two-variable problem because they will offer more flexibility for optimizing future improvements of the model that could potentially involve many more variables. We decided to compare a classical Genetic Algorithm (GA) and a Differential Evolution (DE). For the GA, we use an arithmetic crossover rule (rate=0.6) and the mutation (rate=0.2) was done by adding some random noise to one of the variables. For the DE, we use a differential weight of 1.5 and a crossover probability of 0. For both algorithms, the population size is 50. 100 generations are computed for each run.

3. RESULTS AND CONCLUSION

We considered several scenarios with $n = 10, 20, 30, ...100$ turbines on a square farm of size $3km \times 3km$, and with $n = 200, 300, 400, 500$ and 1000 turbines on rectangular farms of size $8km \times 5km$, $10km \times 6km$, $12km \times 6km$, $14km \times 7km$ and $20km \times 10km$.

Table 1 compares Wagner's results [4] to the patterns optimized with a GA and a DE algorithm on 100 runs for each scenario. The first three columns give the size of the problem, the best and mean energy outputs in kW obtained by Wagner's algorithm (TDA stands for Turbine Distribution Algorithm). Column 4 gives the computation time. Columns 5 and 6 give the best and mean results obtained with the GA for 100 runs. Column 7 gives the mean time for 30 runs (to compare with Wagner's results). Even if the processors used are different, for $n = 1000$, time is divided by 25. Wagner uses 200 000 evaluations, whereas the GA that we implemented required only 4000. Column 8 gives the gain(+) or loss(-) obtained compared to Wagner's results. The next columns give the results obtained with the DE algorithm. For big farms ($n \geq 200$) optimizing the farm pattern is more effective. Surprisingly, this is also true for small farms ($n \leq 50$). For mid-size farms, better results can be found in [4].

To conclude, we have shown that optimizing a simple pattern outperforms existing results on configurations involving a large number of turbines. The gain obtained for farms of 400 and more turbines exceeds 3%, while reducing computation time by an order of magnitude.

By reducing the size of the problem, we focused on optimizing efficiently a small number of variables, and we compensated the loss of generality of the model by a "better" optimization of the remaining variables.

In future research this model can be complexified in order to create more elaborated shapes and to increase the variety of the solutions found. The results can also be used as a starting point for other algorithms requiring a good starting point or starting population.

4. REFERENCES

[1] J. S. Gonzalez, M. B. Payan, J. M. R. Santos, and F. Gonzalez-Longatt. A review and recent developments in the optimal wind-turbine micro-siting problem. In *Renewable and Sustainable Energy Reviews*, volume 30, pages 133–144. Elsevier, 2014.

[2] A. Neubert, A. Shah, and W. Schlez. Maximum yield from symmetrical wind farm layouts. In *Proceedings of DEWEK 2010, the 10th German Wind Energy Conference*, 2010.

[3] H. E. Neustadter and D. A. Spera. Method for evaluating wind turbine wake effects on wind farm performance. *Journal of Solar Energy Engineering*, 107(0):240 – 243, 1985.

[4] M. Wagner, J. Day, and F. Neumann. A fast and effective local search algorithm for optimizing the placement of wind turbines. In *Renewable Energy*, volume 51, pages 64–70. Elsevier, March 2013.

[5] M. Wagner, K. Veeramachaneni, F. Neumann, and U.-M. O. Reilly. Optimizing the layout of 1000 wind turbines. In *European Wind Energy Association Annual Event*, 2011.

[6] F. Wang, D. Liu, and L. Zeng. Study on computational grids in placement of wind turbines using genetic algorithm. In *Proceedings of the World Non-Grid-Connected Wind Power and Energy Conference, 2009*, pages 1–4. IEEE, 2009.

[7] D. Wilson, E. Awa, S. Cussat-Blanc, K. Veeramachaneni, and U.-M. O. Reilly. On learning to generate wind farm layouts. In *GECCO '13: Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, pages 767–774, New York, NY, USA, 2013. ACM.