DEVELOPMENTAL MODEL FOR WIND TURBINE POSITIONING

Emmanuel Awa, Sylvain Cussat-Blanc (Brandeis Demo Lab),
Kalyan Veeramachaneni (MIT Csail EVO Design Group)

Brandeis University
Graduate School of Arts and Science
Department of Computer Science

ABSTRACT

This paper proposes a local method that gives optimizes wind turbine layout compared to already existing global methods. The global existing methods have proved to be inadequate to maximize wind resources towards this goal, hence the need for an optimized algorithm. Our method is based on a bio-inspired developmental model were cells are controlled by a Gene Regulatory Network. We propose to setup a binding between cells and turbines. Since cells can divide, we plan to implement this self-reconfiguring and environmental-adapting approach to turbines. A turbine cell is allowed to adjust to its environment, then it dynamically chooses what direction to split to, that maximizes the wind resource.

***Index Terms*—** Genetic Regulatory Networks, Wind Turbine Optimization, Positon Optimization, Cell-Based Developmental model

1. INTRODUCTION

Wind power is beginning to play a major role in the American portfolio for renewable energy options. Its growth is very fast, about one - fifth of the entire globes capacity of turbines are deployed in America. The need to generate the required power, of about 300 GW resources by 2030, demands effective principles and methods [CSAIL, 2012]. Wind farm design is complex with a lot of issues surrounding maximizing the wind energy. The size of the turbines tends to limit options on how to place them. There are security measures that need to be checked, with certain distances maintained to avoid hazards. Moreover, depending on the wind direction, one turbine can completely refract the wind off its neighbouring turbines. These local interactions between the turbines strongly complicate the optimization of their positioning.

The global existing methods have proved to be inadequate to maximize wind resources towards this goal, hence the need for an optimized algorithm. Global methods involves the use of a grid-structure, putting or not a turbines in each square. This method produces results worse than the solutions proposed by a human engineer, both in quality and in time[kal,].

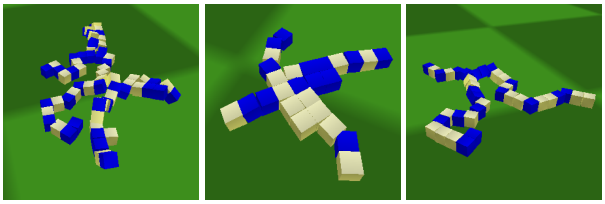


Fig. 1. Diagrams of robot morphologies generated by GRN-based developmental model

Our main project is to use a cell-based developmental model to position wind turbines. The behavior wind turbine cells are controlled using a Genetic Regulatory Network, GRN.

A binding is setup between GRN cells and the wind turbines. Using the way cells evolve in nature, a turbine cell is allowed to adjust to its environment. It then dynamically chooses what action to carry out that maximizes the wind resource. It can decide to divide, wait, rotate or kill itself. In this developmental process, the turbine cell can orient their division plan. If a division is chosen, a child inherits the division plan of the parent and can then take decision on its own. The positive results this developmental model has produced in other fields is the primary propeller of our intention to use it in this project.

This paper is organized as follow. The next section introduces the functioning of a real gene regulatory network. It also details our implementation using the regulatory network. Then, we propose a method on how we plan to implement the regulatory network in wind turbine positioning. Next, we present a set of graphs obtained with our system and animation screen shots of the intended growth of a wind form. Finally, the paper concludes on the future work opened by this approach.

2. GENE REGULATORY NETWORK

Close to two decades, plausible biologically based developmental models have being developed by researchers in the field or artificial embryogenesis. Banzhaf's work on gene

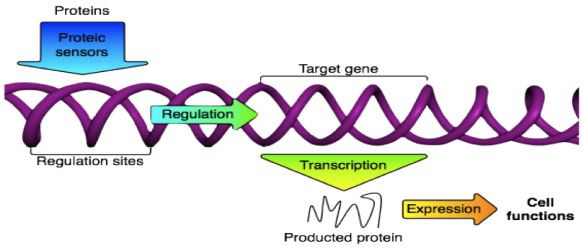


Fig. 2. Concept of a GRN

regulation is the basis of these works [Banzhaf, 2011]. This same cell-based developmental model has being used to generate robot morphologies. It was used in simulating artificial organisms inside a physics simulator by [Pol,]. A graphical representation of this robot morphology is shown in figure 1.

A GRN is a collection of DNA segments in a cell that interact with each other and neighboring cells. This interaction then governs the rate at which genes are transcribed in the network. Cell membranes have protein sensors and these sensors activate or inhibit regulatory sequences of the DNA. Based on the density of the protein at the regulatory sites, a target gene is transcribed into a cell function, resulting to an expression [S. and J., 2012]. Figure 2 is a graphical representation of a GRN.

2.1. Our Simplified Model

Our model is based on Banzhaf's model [Banzhaf, 2011], which closely resembles the real GRN. Due to the complexity of a real GRN in our case, we decided to model it in terms of weighted network of proteins. We chose our "NODES" as concentration weighted proteins. The "EDGES" represented the affinity between two proteins.

The structure of each protein is as follows:

- The protein *identifier* is defined as an integer between 0 and p.
- The *enhancer identifier* is defined as an integer between 0 and p.
- The *inhibiter identifier* is defined as an integer between 0 and p. This is used in calculating the inhibiting matching factor between two proteins.
- The *type* of protein is classified into three. *Input* protein (Its concentration is given by the environment of the GRN. It regulates other proteins but is never regulated), an *output* protein (Its concentration serves as output for the network. It is regulated but does not regulate any other protein) or a *regulatory* protein (This is the internal protein. It regulates and is regulated by other proteins).
- The precision of the network was observed to produce good results at p = 64.

At each time step, we update the concentrations of the proteins. It transforms the dynamics of the GRN and the affinity between proteins a and b for the enhancing factor is computed using u^+_{ab} and the inhibiting u^-_{ab}:

$$u^+_{ab} = p - |enh_a - id_b| \;\; ; \;\; u^-_{ab} = p - |inh_a - id_b|$$

where id_x is the identifier of protein x, enh_x is the enhancer identifier and inh_x is the inhibiting identifier.

The global enhancing value e_i and inhibiting value h_i is then computed using the following equations:

$$e_i = \frac{1}{N}\sum_j^N c_j e^{\beta u^+_{ij} - u^+_{max}} \;\; ; \;\; h_i = \frac{1}{N}\sum_j^N c_j e^{\beta u^-_{ij} - u^-_{max}}$$

where N is the number of proteins in the network, c_j is the concentration of protein j and u^+_{max} (resp. u^-_{max}) is the maximum enhancing (resp. inhibiting) matching factor observed. β is a control parameter.

This differential equation of the concentration of the protein with respect to time, provides the final modification for the model:

$$\frac{dc_i}{dt} = \frac{\delta(e_i - h_i)}{\Phi}$$

Φ is a *Normalization* function for all protein concentrations.

The transitions of the GRN (reaction speeds of the regulatory network) are controlled using two constants β and δ. Where a high value, prompts for a more sudden the transitions and a lower value prompts a smoother transition.

3. DEVELOPMENTAL MODEL USING THE GRN

The GRN can take in many inputs. For the sake of this project we decided to use the Wind Energy distribution as the inputs for the GRN cells. The distribution is the output values gotten from the already existing wind turbine simulator.

Based on the genetic algorithm, the GRN can perform one of eleven possible outputs which are as follows:

- It can divide according to its division plan.
- The cell can decide to rotate based on its division plan alignment to 8 cardinal directions (North, North-East, East, South-East, South, South-West, North-West and West).
- The cell can decide to kill itself (Apoptosis).
- The cell can decide to wait.

Our experiment is based on simulations for now. A land is initially mapped out as a grid and then our developmental model is created all over the grid. A GRN is placed in any arbitrary cell location and allowed to dynamically evolve. The evolution of the cell allows it to dynamically make one decision by adapting and observing its environment. The cell at some point will decide to divide in the direction of its division plan. The child cell completely inherits all the attributes of the parent and itself becomes an individual cell that makes its own decision. The difference right now, is that consideration is given to other cells around it. Cells can only divide in the direction of their division plan and cannot overlap each other. If a GRN exists in the cell where another GRN wants to divide to, it automatically makes another decision either to wait, rotate or kill itself.

We currently have the following functions of our developmental model:

- DevGrid.m - The input for this function is the required size of cells a farm will have. It then creates a developmental structure, setting all cells as dormant and facing North.
- addCell(LocationIndices).m - This function activates the cell location that has being passed as inputs. If the cell is already activated, it prompts an error and terminates.
- stepForward.m - This function takes a developmental grid as input and outputs an updated developmental grid. It checks each cells over a number of time steps and determines what action to take. The action is ultimately triggered by the presence of a cell that is alive. For now, the actions and decisions are random.

Our developmental model is a cell array of structures and our default representation of a cell (at the initialization of a developmental model) is as follows:

$$cellValue = struct(); \tag{1}$$

$$cellValue.Alive = boolean(0); \tag{2}$$

$$cellValue.Direction =' North'; \tag{3}$$

Figure 3 shows the simulation of the developmental model from placing one cell to the division plan across two other cells.

4. GRN BEHAVIOR AND RESULTS

To achieve the best result we had to test and make sure our GRN works for the purpose we intend to use it. From Banzhafs model [Banzhaf, 2011], the GRN is expected to display certain behaviors. These behaviors are Chaotic, Transitory and Stable.

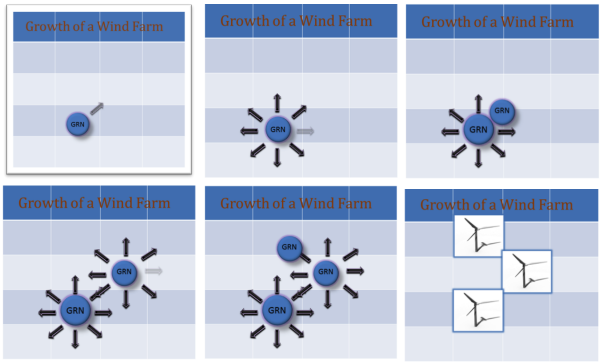


Fig. 3. Simulation diagram of the developmental model in the growth of a wind farm

We made a plot of the protein concentration over time steps and our results showed that we had a fully functional GRN.

Figure 4 (a) shows a chaotic behavior of the GRN. All the cells are oscillating at the same time showing little or no control in the environment.

Figure 4 (b) shows a transitory behavior. Such expression is observed with the cells oscillate one after another. Then Figure 4 (c) shows the most stable state of the GRN. It can be observed that stability was achieved at about 200 step forwards in the evolution. It matches the results gotten by Banhzaf (Banhzaf, 2003) for his GRN model, showing our model works.

4.1. Optimization of the GRN

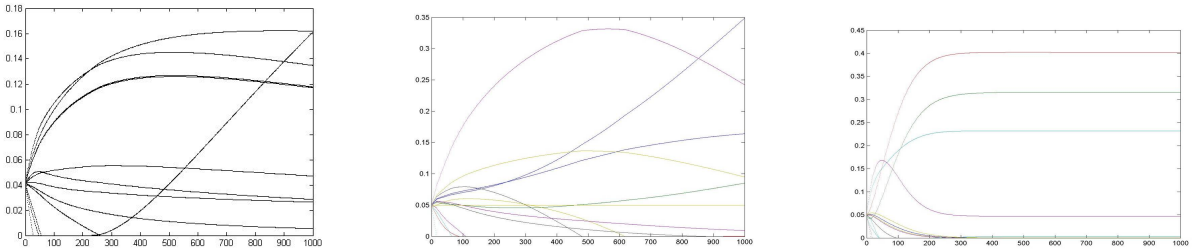
We are currently building the developmental model using random inputs and have observed the need to optimize some interesting features. The GRN needs to be evolved to optimize the cell behavior in other to get interesting results. We found out that evoluting the GRN below 1000 steps did not give us interesting behaviors.

Evolving the GRN means we need to encode the GRN in a genome and genetically evolve it. Then we intend to evaluate this genome to get the fitness function.SCB.

The possible fitness's could be one of the following:

- Direction with most energy can be obtained as output from our model.
- Energy loss due to neighbors can be easily obtained and based on that, optimization judgments can be made.
- Possible inputs for other GRN's can be obtained. A classical example is the case where a GRN decides to split.
- Turbines can be effectively positioned after a stable state is observed using our model in the wind turbine simulator.

Fig 4 (a) Fig 4 (b) Fig 4 (c)



Chaotic behavior Transitory behavior Stable behavior

Fig. 4. Example of behaviors generated by a GRN. Plot of protein concentration against time steps.

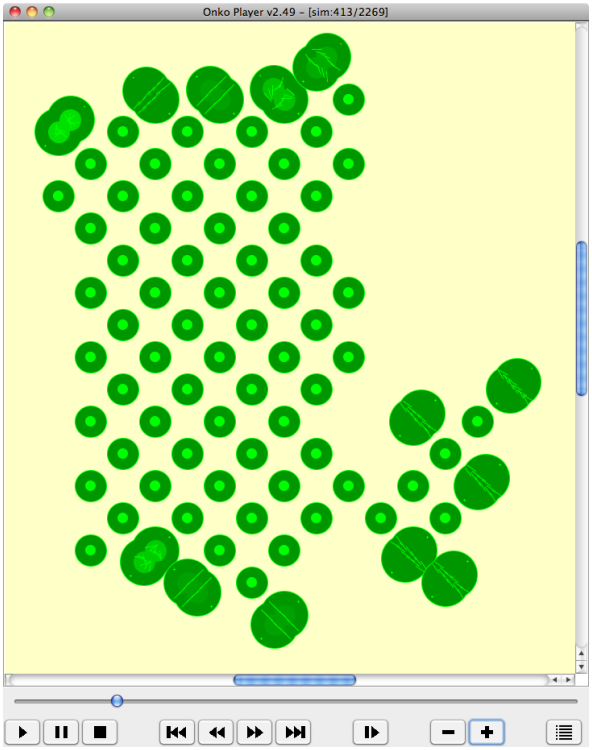


Fig. 5. Expected layout output

5. CONCLUSION AND PERSPECTIVE

In this paper we have proposed a developmental model for wind turbine positioning. We coded our GRN in matlab and using random variables, we were able to ascertain that our GRN works and can be used in the turbine positioning. We hope to finish building the developmental model and then interfacing it with the Wind turbine simulator. Based on our computation and experiments, we expect to get great results for layout optimization. Using a GUI interface developed by students in France [Tournois et al., 2011], we expect to obtain an output that looks like Fig 5 bar.

We expect our local method would help decide how to place these turbines on the fields.

6. FUTURE WORKS

For the fact that our project is on-going, we do not envisage so much future work beyond the primary goal. They are as follows:

- We intend to optimize our GRN to accept different types of inputs and solve the same layout problem.
- Complete coding the developmental model, interface it and obtain accurate results.
- Use other approaches like Genetic programming or Neural networks to control the cells.

7. REFERENCES

[Pol,] A Cell-based Developmental Model to Generate Real Robot Morphologies. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, GECCO '2012.*

[kal,] Optimizing the layout of 1000 turbines. In *Proceedings of Annual European Wind Energy Conference (EWEA), 2011.*

[Banzhaf, 2011] Banzhaf, W. (2011). Artificial Regulatory Networks and Genetic Programming.

[CSAIL, 2012] CSAIL, E.-D. (2012). Energy Information Technology Website. http://groups.csail.mit.edu/EVO-DesignOpt/evo.php?n=Site.WindEnergySystems/. [Online; accessed 20-February-2012].

[S. and J., 2012] S., C.-B. and J., P. (2012). Using pictures to visualize the complexity of gene regulatory networks.

[Tournois et al., 2011] Tournois, S., Cussat-Blanc, S., Pascalie, J., Luga, H., and Duthen, Y. (2011). An open-source generic rendering tool for developmental models.