Deploying advanced optimization algorithms – a case study

How Ørsted leverages cutting-edge decision support tools to create a world that runs entirely on green energy

Ørsted

Richard Oberdieck
07/05/2019

Ørsted at a glance

Headquartered in Denmark
Listed in the Nasdaq OMX: ORSTED
5,638 employees
Revenue in 2017 DKK 59.5 bn (GBP 7.1 bn / EUR 8.0 bn)
EBITDA in 2017 DKK 22.5 bn (GBP 2.7 bn / EUR 3.0 bn)
Phase out the use of coal by 2023

83%* Wind Power

- Develops, constructs, owns and operates offshore wind farms in Denmark, Germany, the Netherlands and the UK
- Development projects in Taiwan and the USA

4%* Bioenergy & Thermal Power

- Generates and sells power and heat to customers in Denmark and Northwestern Europe

13%* Distribution & Customer Solutions

- Power distribution grid on Zealand and sale of power and gas to customers in Northwestern Europe

Share of the Ørsted Group's capital employed, Annual Report 2017

A little bit about myself

Education:

- B.Sc. and M.Sc. in Chemical Engineering from ETH Zurich
- Ph.D. in Process System Engineering from Imperial College London

Work experience:

- Since Feb. 2017 with Ørsted as a Numerical Specialist
- Guest lecturer at DTU on practical optimization tools & techniques

Personal:

- Married, 1 son
- Enjoy basketball, camping and BBQ-ing

When deploying advanced decision support systems, we have learned 4 key lessons to help us make the tool a success

Focus on users

- **The closer** you work with the users, **the better**
- **Show your progress**, even when incomplete
- Be **computational conscience** (= “Do we really need this?”)
- **Agree** at every step of the way

Test, test, test

- **Unit test** are your single best friend in math tools
- Look at the **mathematical edge cases**
- It is **very (!) difficult** to find **math mistakes** such as scaling factors etc.

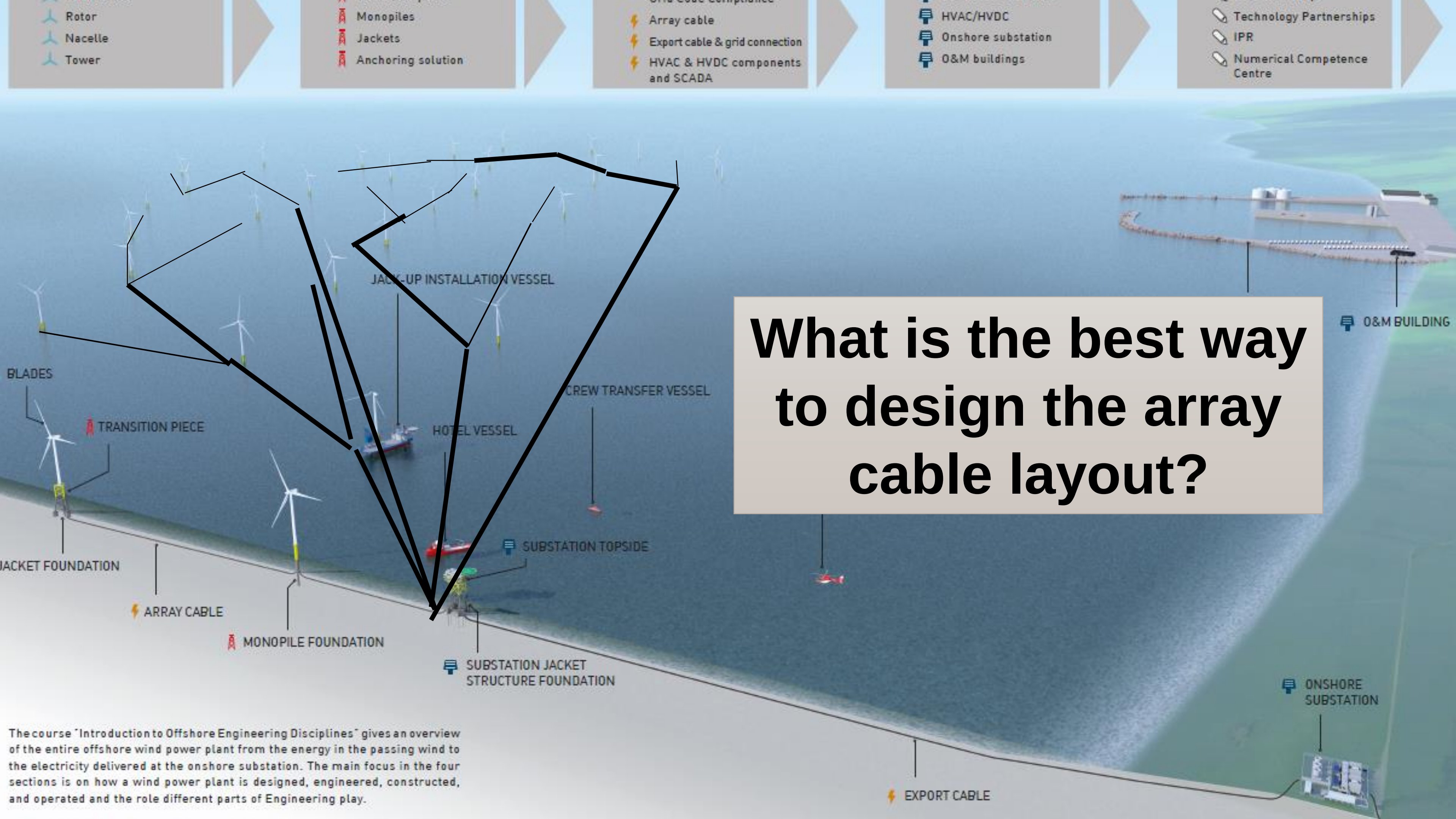
Document it

- **Keep a separate document** that shows your ideas behind the model/algorithm
- Very **few people** can **check** your work
- **Share your documents** with others for feedback

The UI is key

- People **equate tools** with their **interfaces**
- **The simpler, the better** (e.g. Google)
- Interfaces should be “**respectful**, **generous** and **helpful**”

Ørsted


Rotor
Nacelle
Tower
Monopiles
Jackets
Anchoring solution
Array cable
Export cable & grid connection
HVAC & HVDC components and SCADA
HVAC/HVDC
Onshore substation
O&M buildings
Technology Partnerships
IPR
Numerical Competence Centre
What is the best way to design the array cable layout?
JACK-UP INSTALLATION VESSEL
O&M BUILDING
BLADES
TRANSITION PIECE
CREW TRANSFER VESSEL
HOTEL VESSEL
SUBSTATION TOPSIDE
JACKET FOUNDATION
ARRAY CABLE
MONOPILE FOUNDATION
SUBSTATION JACKET STRUCTURE FOUNDATION
ONSHORE SUBSTATION
EXPORT CABLE
The course "Introduction to Offshore Engineering Disciplines" gives an overview of the entire offshore wind power plant from the energy in the passing wind to the electricity delivered at the onshore substation. The main focus in the four sections is on how a wind power plant is designed, engineered, constructed, and operated and the role different parts of Engineering play.

In a more abstract way…

Given a set of turbines and offshore substations, find the best way to connect all turbines to the substations such that cost is minimized while respecting a variety of constraints

minimize	CAPEX + OPEX
subject to	All turbines are connected
	No cables are allowed to cross
	Limit number of turbines on a given string
	Limit number of ingoing connections to turbine and OSS
	Choose from a variety of cables
	CAPEX = Cables + Connections
	OPEX = Power losses

- This used to be done manually
- However, starting with two M.Sc. students from DTU in 2012/13, we started developing a dedicated software tool to solve this optimization problem: OptiArray
- Since then, we have been continuously improving it to be more user-friendly and consider a greater variety of situations.

Let's look at the rules in action for OptiArray, our tool for the design of array cable layouts in offshore wind parks

Focus on users:

- We ran the development in sprints, with daily meetings with the two key users.
- Every aspect (even the math) was explained on a high level to users.
- The users (and only the users) can accept what you have developed.

Test, test, test:

- Every part of the code has unit tests associated with it
- Unit testing of optimization code is difficult, but is always worth it.
- Remember the difference between unit tests and integration tests.

Document it:

- We wrote a confidential document that "spills out" all the details of the model and heuristics used.
- It is referenced to from the code
- Difficult to keep up to date; there is no good tool to do this.

The UI is key:

- Most of the user comments refer to the UI
- The math is complex, so rarely challenged
- By binding in the users early, they cannot say "I don't like this UI"

Ørsted

Conclusions

Advanced decision support systems are a key to success, but only when done right.

Always keep the value added for the user in the focus, rather than the beauty of the math.

Robustness is key: tools are supposed to be used over and over, not one time only.

Don’t take short cuts. In the end, it will almost always bite you.