THE UNIVERSITY OF NEWCASTLE AUSTRALIA

SCHULICH School of Engineering UNIVERSITY OF CALGARY

Blade Arrangement for Multi-blade Rotors

David Wood
Konrad Hitz

24 ft Kijito windmill in northern Kenya. Original design from ITDG

Water pumping windmills can have up to 24 blades:

- Blades are often made only as required for each machine
- Potential for significant imbalance
- Impossible to optimally minimize the imbalance in finite computer time
- Randomness should help
- No proper framework to address the problem

Background

Number of blades on wind turbine or windmill = *N*
Number of blades made in one batch = *B*

For turbines often *B* >> *N*
where *N* = 2 or 3 and the problem is blade matching – see Hitz & Wood (2010)

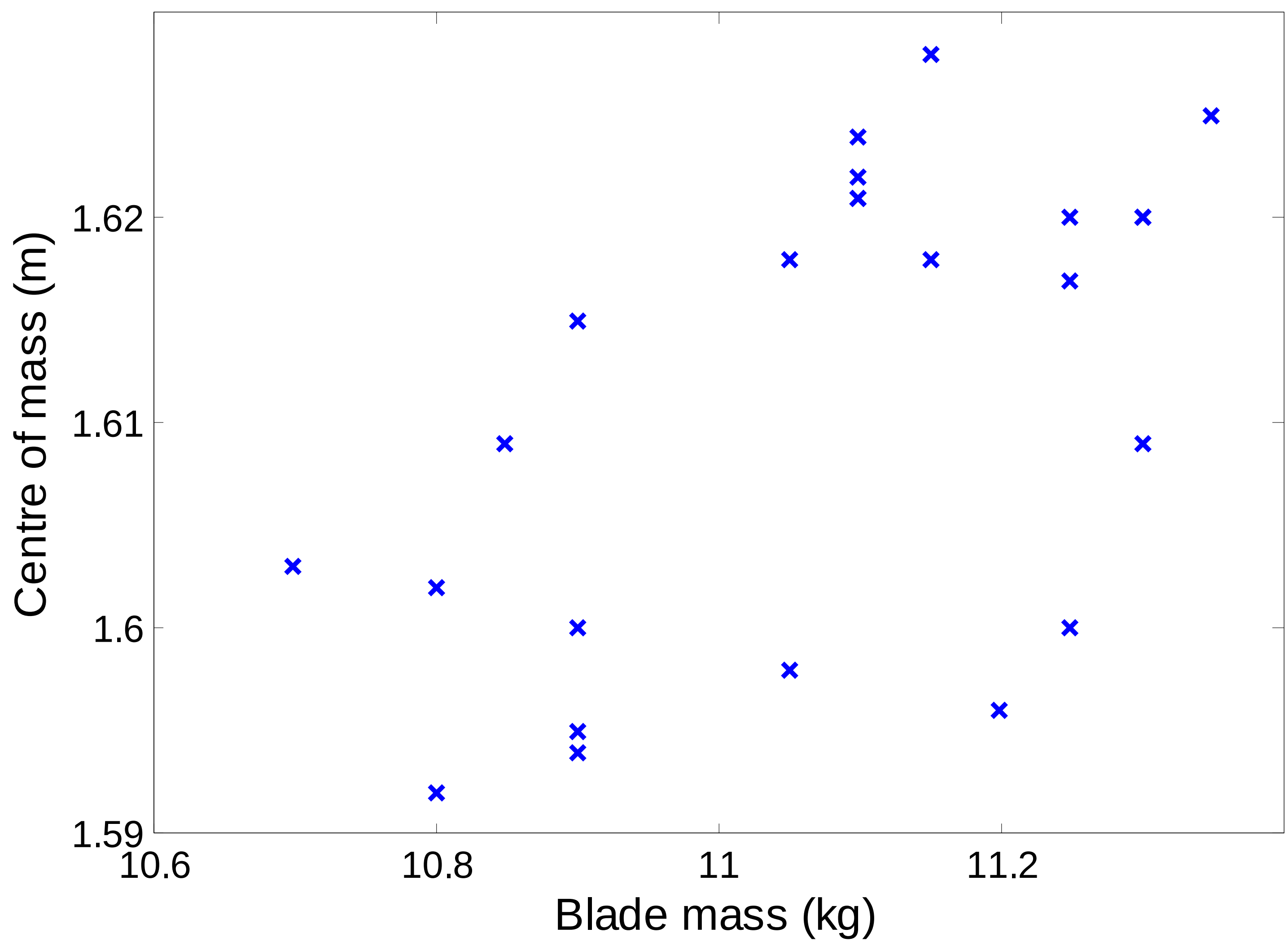
For windmills *B* = *N* for large *N* and the problem is blade arrangement

Measurements of blade mass and centre of mass at Kijito Windpower in Kenya

Blade Number	Blade Mass (kg)	Centre of Mass (m)	Blade Number	Blade Mass (kg)	Centre of Mass (m)
1	11.3	1.62	13	10.7	1.603
2	10.9	1.6	14	11.25	1.6
3	11.3	1.62	15	11.25	1.617
4	10.9	1.615	16	11.1	1.621
5	11.15	1.628	17	10.85	1.609
6	11.05	1.598	18	11.25	1.62
7	10.8	1.602	19	10.9	1.594
8	11.2	1.596	20	11.3	1.609
9	10.9	1.595	21	10.8	1.592
10	11.15	1.618	22	11.05	1.618
11	11.35	1.625	23	11.1	1.621
12	11.1	1.624	24	11.1	1.622

Blade numbering denotes order of measurement. It has no other significance

Correlation coefficient between mass and centre of mass = 0.55



IEC 61400-2 for small wind turbines assumes a default eccentricity of the rotor centre of mass, *e*, of *e* = 0.005*R*, where *R* is the blade radius

d_l is the product of the blade mass and centre of mass for blade *l*. The vector sum of *d* for the rotor can be written as

$$\mathbf{d} = \sum_1^N d_l \cos(2\pi l / N)\mathbf{i} + \sum_1^N d_l \sin(2\pi l / N)\mathbf{j}$$

e is given by

$$e = \sqrt{d^2} \Big/ \sum_1^N m_l \quad \text{where} \quad d^2 = \sum_1^N \sum_1^N a_{ml} d_m d_l$$

$$\text{and} \quad a_{ml} = \cos(2\pi(l - m)/N)$$

Note that $a_{ll} = 1$

For $N = 2$, $a_{12} = a_{21} = -1$ and for $N = 3$ all the off-diagonal elements in a_{lm} are equal to -1/2. Thus the arrangement of blades in either case does not affect the eccentricity. For $N = 4$, the off-diagonal elements are either -1 or 0, so swapping within the two pairs of blades, say (1,3) and (2,4), does not change e but any other swap will. For $N > 4$, a_{lm} changes sign, so the blade arrangement is critical.

Strategies for Blade Arrangement given:

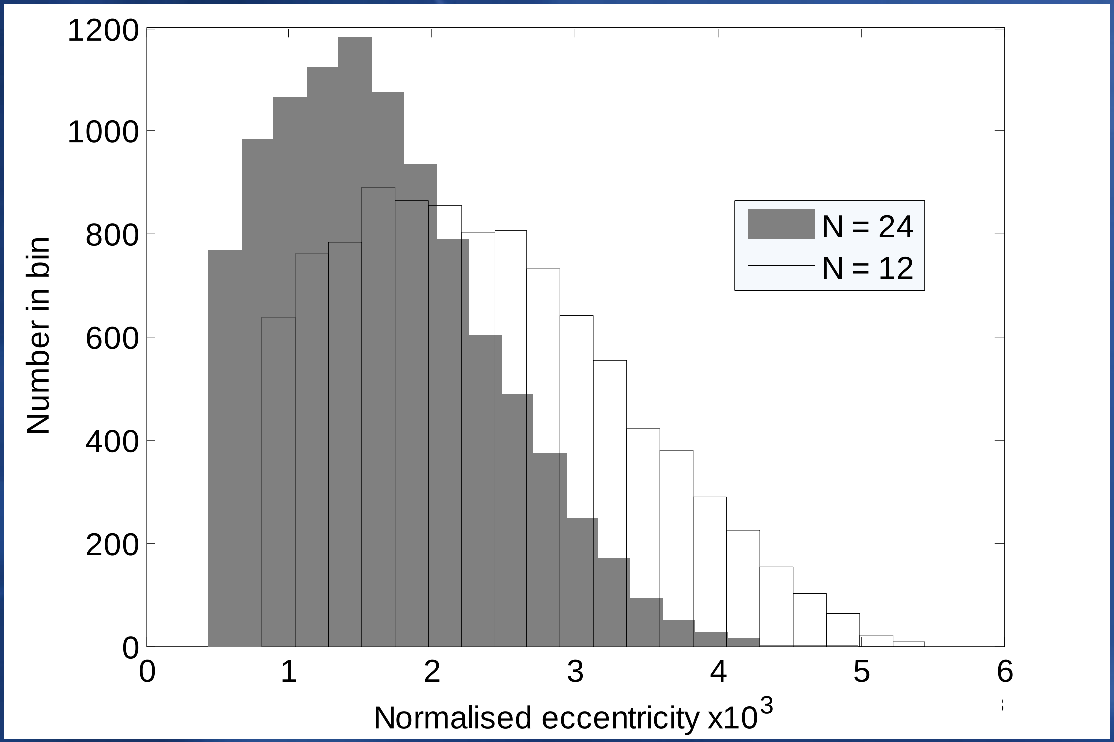
1. $24! = 6.2045 \times 10^{23}$
2. Branch and bound algorithms that Hitz & Wood (2010) developed for blade matching will not work as a_{lm} can be positive or negative

Strategies Investigated:

1. Random selection
2. Order blades into pairs or triples and then select randomly
3. Develop a heuristic based on pairing

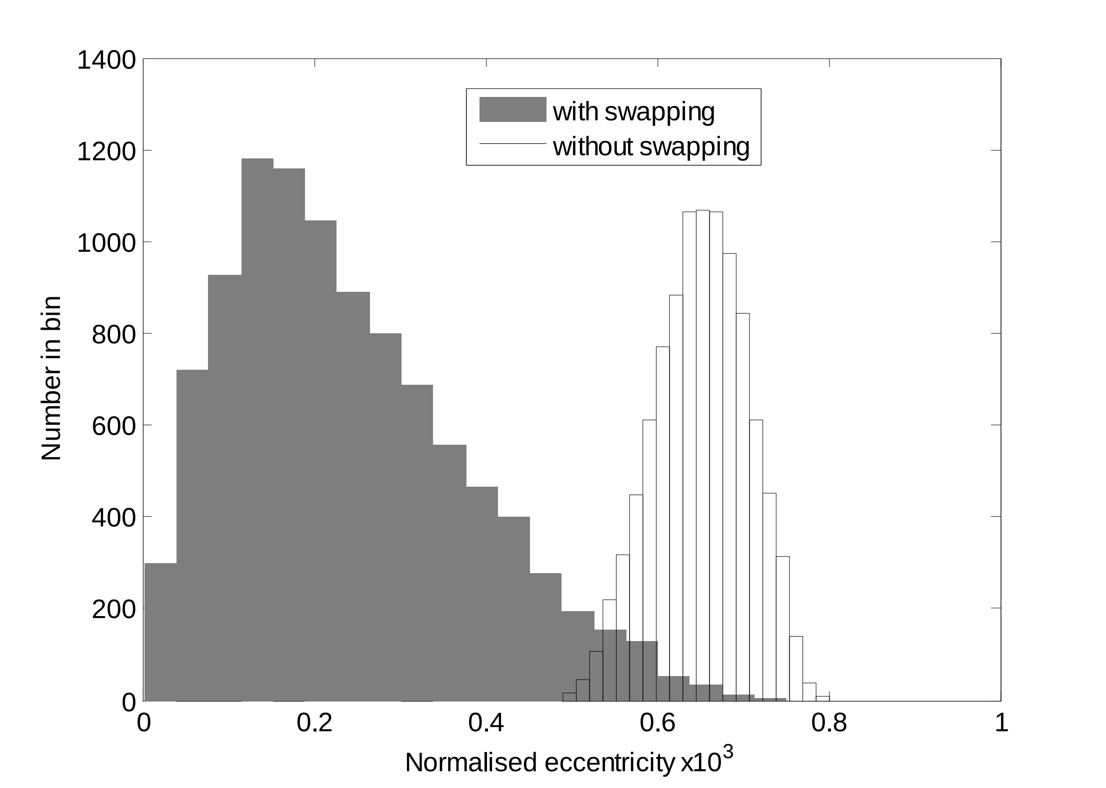
1. Random selection

Histograms show 10,000 random selections of blades for 12-blade and 24-blade rotor



Normalised eccentricity is *e/R*, where *R* is the blade radius

2. Random selection after pairing for N = 24. Pairing is based on ordering in d



Histograms show 10,000 random selections of blades for a 24-blade rotor

Pairing significantly reduces e

Swapping every second blade pair is even better

Swapping reduces the eccentricity because d^2 can be written as

$$d^2 = \sum_{1}^{N} \delta_l^2 - \sum_{l=1}^{N} \delta_l \delta_{l+N/2} + \sum_{m=l+1}^{N} \sum_{l=1}^{N} a_{lm} \left(\delta_l - \delta_{l+N/2}\right)\left(\delta_m - \delta_{m+N/2}\right)$$

where δ is the deviation from the average

Heuristic

1. Pair blades
2. Assign blades 1,2 to an arbitrary rotor diagonal
3. Search for the best position pair for blades 3,4 from the remaining unoccupied rotor diagonals; for each diagonal, check whether the assignment should be 3,4 or 4,3.

Repeat step 3 for successive blade pairs {(5,6),…(N-1,N)} until the assignment is complete.

	Normalised Eccentricity × 10^3 (min,mean,max)		CPU secs	
N	Heuristic	Optimal	Heuristic	Optimal
6	(0.246, 0.987, 2.584)	(0.126, 0.615, 2.153)	0.0001	0.028
8	(0.075, 0.552, 1.516)	(0.008, 0.083, 0.287)	0.0001	0.109
10	(0.037, 0.361, 1.120)	(0.001, 0.011, 0.035)	0.0001	7.92
12	(0.043, 0.355, 0.710)	(0.000, 0.001, 0.002)	0.0001	874.0
24	0.019	-	0.0078	-

Heuristic is fast, gives low values of *e* but not as low as the optimal value when this could be found

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

SCHULICH School of Engineering UNIVERSITY OF CALGARY

Conclusions

1. Random selection is a good strategy as *N* increases
2. Heuristic is a good strategy
3. Pairing, swapping and random assignment of blade pairs also works well

Thank You