대한산업공학회/한국경영과학회 2000 춘계공동학술대회 논문집. 2000년 4월 21-22일. 경남대학교
Session OB2.5

터빈 블레이드 균형화를 위한 발견적 기법
A Heuristic Algorithm for a Turbine-Blade-Balancing Problem

최원준
울산대학교 공과대학 산업공학부 choiwj@uou.ulsan.ac.kr

Abstract

In the turbine-blade manufacturing industry, turbine-blades are machined and then are assembled to form a circular roll of blades. The roll of blades should be balanced as much as possible, since otherwise the efficiency of the turbine generator might degrade. We propose a heuristic method for balancing blades based on the number partitioning algorithm. The proposed method outperformed existing methods remarkably in terms of the accuracy with a negligible increase in the running time.

1. INTRODUCTION

A steam turbine may be defined as a form of the heat engine in which the energy of the steam is transformed into kinetic energy by means of expansion through nozzles, and the kinetic energy of the resulting jet is in turn converted into force doing work on rings of blades mounted on a rotating part.

In the turbine-blade manufacturing industry, turbine-blades are machined and then are assembled to form a circular roll of blades. The roll of blades should be balanced as much as possible, since otherwise the efficiency of the turbine generator might be damaged. However, the blades to be assembled into the same roll are not normally identical in weights and lengths, which makes the balancing problem tedious and difficult.

In this paper, we study methods for balancing the weights of the rotating blades. In Section 2, we formulate the blade-balancing problem into a mixed-integer programming problem. In Section 3, we review the literature on topics of the turbine blade balancing problem. In Section 4, we propose a heuristic method for solving the blade-balancing problem, followed by the exposition of the computational experiences in Section 5.

2. PROBLEM-FORMULATION

Suppose we are given a set of n blades whose weights are known. The problem which we should solve is to determine the location of each blade around the rotor axis so as to minimize the *residual unbalance* (its definition will be given later) in weight distribution. For expository simplicity, we assume that the centers of gravity of all blades are at the same distance r from the center of the rotor axis. We define the following notation:

n: total number of blades, equivalently total number of locations
i: blade index (i=1,2,...,n)
j: location index (j=1,2,...,n)
w_i: weight of blade i.
r: distance between the center of gravity of a blade and the center of rotor axis.

Then the balancing problem can be defined as follows: Given n blades with weight w_i and a circle of radius r with n equally spaced locations on its periphery, find an assignment of the blades to the locations that minimizes residual unbalance about the center. The residual unbalance is the magnitude of the vector sum of the moments created by the individual blades about the center.

Without loss of generality, we assume that n is even. For convenience, we assume a coordinate system in which the origin is at the center of the circle and the positive x axis goes through one of the n locations as in Figure 1. The locations are numbered in counterclockwise order starting with the one coincident with the positive x axis. So, the coordinates of location i are

$$\left(r\cos\left(\frac{2(i-1)\pi}{n}\right), r\sin\left(\frac{2(i-1)\pi}{n}\right)\right), \quad i = 1,\dots,n.$$

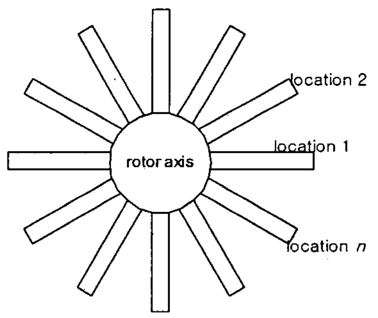


Figure 1. Blade Balancing Problem

Decision variables x_{ij} are defined as

$$x_{ij} = \begin{cases} 1 & \text{if blade } i \text{ is assigned to location } j \\ 0 & \text{otherwise} \end{cases}$$

Let

$$c_{ij} = w_i \bullet r\cos\left(\frac{2(i-1)\pi}{n}\right),\quad d_{ij} = w_i \bullet r\sin\left(\frac{2(i-1)\pi}{n}\right)$$

.Now, for any solution x we can determine the moment vector (m_x,m_y) of the moment weight as

$$m_x = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \qquad \text{(Eq. 1)}$$

$$m_y = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} \qquad \text{(Eq. 2)}$$

Then the balancing problem can be stated as follows:

$$Minimize\ \sqrt{m_x^2 + m_y^2} \qquad \text{(Eq. 3)}$$

subject to

$$m_x = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \qquad \text{(Eq. 4)}$$

$$m_y = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} \qquad \text{(Eq. 5)}$$

$$\sum_{i=1}^{n} x_{ij} = 1,\quad j = 1,2,\ldots,n, \qquad \text{(Eq. 6)}$$

$$\sum_{j=1}^{n} x_{ij} = 1,\quad i = 1,2,\ldots,n, \qquad \text{(Eq. 7)}$$

$$x_{ij} = 0,1 \qquad \text{(Eq. 8).}$$

Equations 6 & 7 are assignment constraints for blades and locations. Equations 4 & 5 are definitions of moments.

3. LITERATURE REVIEW

The turbine blade balancing problem has been studied by several authors. Mosevich (1986) presented an algorithm which consisted of selecting the best of a large number of randomly generated solutions. Laporte and Mercure (1988) modeled the problem as a quadratic assignment problem and presented a solution procedure based on Or's TSP heuristic which outperformed that of Mosevich. Fathi and Ginjupalli (1993) also modeled the problem as a quadratic assignment problem and two families of heuristics for it. The first family of heuristics called the Placement Heuristic places the blades in order of weight – the heaviest blade first – choosing for each blade the available location that brings the resulting center of gravity as close as possible to the center of rotor axis. The second family of heuristics called the Rotational Heuristic divides the blades into equal-sized subsets, finds good sequences for the smaller problems of balancing with only the blades in each subset and then interleaves the sequences. Mason and Rönnqvist (1997) tested several local search techniques including pairwise interchange and three-way interchange algorithms. They found that pairwise interchange heuristic was most efficient.

Amiouny, Bartholdi and Vande Vate (1997) developed several constructive heuristics for the problem of which two seem to dominate, Ordinal Pairing and Greedy Pairing Both algorithms begin by sorting the blades from heaviest to lightest. Next after forming the pairs of consecutive blades in the sorted list, sort the pairs in the descending order of the difference in weights in a pair. Then pairs in the finally sorted list are placed across from each other on the rotor axis. The two algorithms differ as to how the locations of each pair of blades are determined. Ordinal pairing places the blades in a fixed pattern. Greedy pairing places the blades by a greedy algorithm. For each pair, all possible open positions on the circle are examined, and the position which yields the center of gravity closest to the center of the circle is chosen. Ordinal Pairing requires very little computation time and produces good results while Greedy Pairing requires more computation but gives better performance.

Choi *et al.* (1999) presented several heuristic methods for the turbine blade balancing problem formulated as a minimax unbalance problem.

Storer(1999) proposed a heuristic which uses an embedded number partitioning algorithm. Initially, the blades are placed in random locations on the circle. Next the center of gravity around "the X-axis" is balanced, and then around "the Y-axis." To balance the center of mass around an axis, pairs of weights which are symmetric with respect to the axis as shown in Figure 2 are considered. Next a single number d_i for each pair of weights is created (i=1 to $n/2$), as illustrated in Figure 3. This number d_i is the center of gravity of the pair with respect to the axis of symmetry. Next a number partitioning algorithm is applied to the set $\{d_1, d_2, \ldots, d_{n/2}\}$ of the pairwise center of mass. The differencing algorithm proposed by Karmarkar and Karp (1982) was applied to solve the number partitioning problem. The result of partitioning is two sets of pairs. The final step is to arbitrarily select one of the two sets of pairs, and interchange the weights of each pair in that set. The result will be that the center of gravity with respect to the axis of symmetry will be nearly balanced and equal to the objective function found by the *number partitioning algorithm.* The final step of the algorithm is to balance the center of gravity with respect to the second perpendicular (Y) axis of symmetry. The same algorithm as for the X-axis is applied.

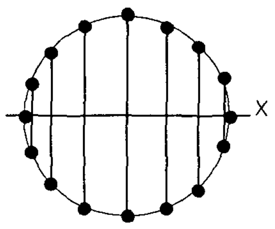
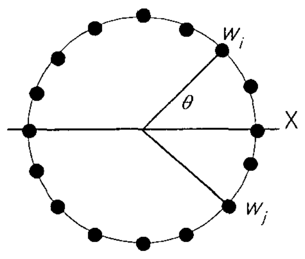


Figure 2 Weight Pairs Symmetric with Respect to the X Axis in Storer's Algorithm



d=|w$_i$-w$_j$|*sin(θ)

Figure 3 Calculation of d values

We will present a heuristic which is based on Storer's heuristic.

4. HEURSITICS

The proposed algorithm begins with an arbitrary placement of blades with centroid (W_x,W_y), and improves the solution iteratively.

While Storer's algorithm first balances the center of gravity around "the X-axis", we deliberately select the axis around which the center of gravity will be balanced. The axis will be chosen among axes with angles

$$\frac{(i-1)\pi}{n}, \quad i=1,\ldots,n.$$

(We will call the axis with angle $\frac{(i-1)\pi}{n}$ by Axis i, for i=1,...,n or say the axis number is i.) We choose the axis that is nearest to the separating line perpendicular to the line segment linking the center (0,0) and the centroid (W_x, W_y) as illustrated in Figure 4.

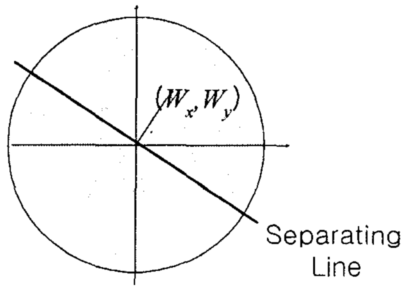


Figure 4 The separating line perpendicular to the line segment (0,0)-(W_x,W_y).

To balance the center of mass around the selected axis, pairs of weights that are symmetric with respect to the axis as shown in Figures 5 and 6 are considered. When the axis number is even, the axis passes between two pairs, but when the axis number is odd, the axis lies on top of two blade locations.

First, each symmetric pair is arranged so that the heavier weight in each pair is on the same side of the axis (for simplicity of the algorithm description). Next a single number d_i for each pair of weights is created. This number d_i is the center of gravity of the pair with respect to the axis of symmetry. That is,

d_i = difference in weights of the pair X sin(difference in angles of the pair / 2).

Next a number partitioning algorithm is applied to the set {d_i} of the pairwise centers of mass. The result of partitioning is two sets of pairs. The final step is to arbitrarily select one of the two sets of pairs, and interchange the weights of each pair in that set. The result will be that the center of gravity with respect to the axis of symmetry will be nearly balanced and equal to the objective function found by the number partitioning algorithm.

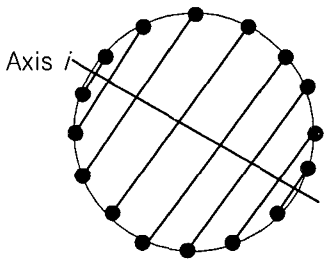


Figure 5 Pairing in Case of Axis i (i is even)

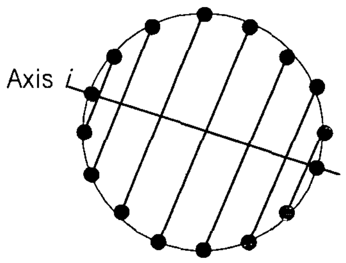


Figure 6 Pairing in Case of Axis i (i is odd)

The rationale behind the selection of the axis is that the nearer the axis is to the perpendicular separating line, the larger the unbalance of the existing placement with respect to the axis is, so that improvement from the existing placement may be more likely.

After applying the above process, we get a new placement and update the centroid of the placement. Then we select the axis around which the center of gravity will be balanced. If the solution was improved and the newly selected axis is different from the previous axis, then we repeat the above process. We call the newly selected axis the anchor axis. Initially the anchor axis is the first selected axis. Otherwise, we explore other pairings by selecting an axis next to the current axis and then repeat the above process. If we fail again to improve the solution, then we try another untried axis nearest to the anchor axis. The algorithm stops when all axes are tried but the anchor axis is not changed.

We call this proposed algorithm the Iterative Method.

5. COMPUTATIONAL EXPERIENCES

Following Amiouny, Bartholdi, and Vande Vate (1997), we generated blade weights from a Normal distribution with a mean of 100 and standard deviation of 5/3. We generated problems over a range of sizes from 20 blades to 200 blades. Again following Amiouny, Bartholdi, and Vande Vate (1997), we assume that the circle radius is 100 and the objective function is the Euclidean distance between the center of gravity and the center of the circle. For each problem size, 1000 instances were generated. Four algorithms were compared, Ordinal Pairing, Pairwise Interchange Method, Storer's Method, and the Iterative Method.

Figure 7 shows the objective function value averaged over 1,000 problem instances for each problem size. Storer's Method outperformed the Ordinal Pairing Method and 2-Swap Method for most of cases. Considering the fact that the Pairwise Interchange Method was one of the best methods in the open literature (Mason and Rönnqvist (1997)) before Storer's Method was proposed, we can say that Storer's Method works very nicely for the turbine blade balancing problem. However, the Iterative Method improves on Storer's Method by up to three orders of magnitude with a negligible increase in the running time in the practical sense. For N=200, the Iterative Method took an average of 0.16 seconds per problem instance on a PC with Pentium II processor.

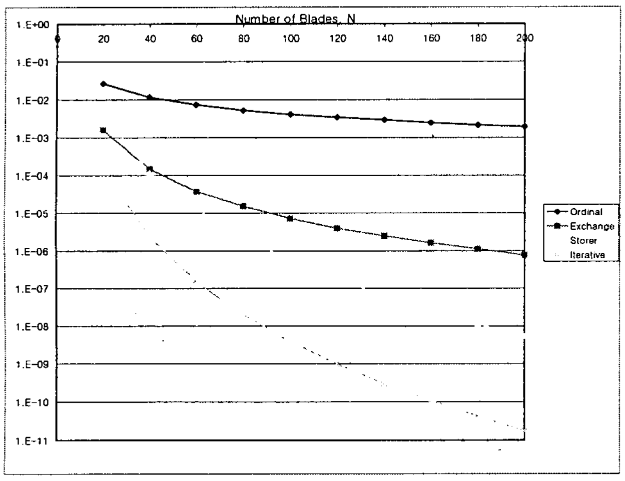


Figure 7 Comparison of the Performance of Ordinal Pairing, Pairwise Interchange, Storer's, and Iterative Method.

Another point to note is that the quality of the final solution from the Iterative Method was robust with respect to the starting solution. We tried using solutions from Ordinal Pairing, and from the Interchange Method as the starting placement. However, there was no indication that the resulting final solutions are better than those from arbitrary starting placements.

6. CONCLUSIONS

In this paper, we proposed an algorithm for a turbine blade balancing problem. It turned out that the proposed algorithm improved on the state-of-art method by up to three orders of magnitude with a negligible increase in the running time.

Investigation of other pairing schemes and other search heuristics such as a problem-space search heuristic remains as further research topics.

REFERENCES

Amiouny, S.V., Bartholdi,III,J.J., and Vande Vate, J.H. (1997), "Heuristics for Balancing Turbine Fans," Technical Report, Department of Industrial and Systems Engineering, The Georgia Institute of Technology, Atlanta, Gergia (Forthcoming in Operations Research).

Choi, W., Kang, H., and Baek, T. (1999), "Turbine Blade Balancing Problems", *International Journal of Production Economics*, Vol. 60-61,405 - 410.

Darlow, M.S. (1989), *Balancing of High-Speed Machinery*, Springer-Verlag, New York.

Fathi, Y. and Ginjupalli, K.K. (1993), "A Mathematical Model and a Heuristic for the Turbine Balancing Problem," *European Journal of Operations Research*, Vol. 63, pp.336-342.

Karmakar, N.R.M. and Karp, R.M. (1982), "The Differencing Method for Set Partitioning," Report No. UCB/CSD 82/113, Computer Science Division, University of California, Berkley.

LaPorte, G. and Mercure, H. (1988), "Balancing Hydraulic Turbine Runners: A Quadratic Assignment Problem.," *European Journal of Operations Research*, Vol. ??, 378-381

Mason, A. and Rönnqvist, M. (1997), "Solution Methods for the Balancing of Jet Turbines," *Computers and Operations Research*, Vol. 24, No. 2, 153-167.

Mosevich, J. (1986), "Balancing Hydraulic Turbine Runners: A Discrete Combinatorial Optimization," *European Journal of Operations Research*, Vol. 26, 202-204.

Papadimitriou,C.H. and Steiglitz,K.(1982), *Combinatorial Optimization*, Prentice-Hall, Inc.

Storer, R.H. (1999), "Extenstions of and Uses for the Differencing Algorithm for Number Partitioning," Report No. 99T-09, Department of Industrial and Manufacturing Systems Engineering, Lehigh University, Bethlehem, Pennsylvania.

Storer, R.H., Flanders, S.W., and Wu, S.D. (1996), "Problem Space Search for Number Partitioning," *Annals of Operations Research*, Vol. 10, 465-487.