European Journal of Operational Research 63 (1993) 336–342
North-Holland

Theory and Methodology

A mathematical model and a heuristic procedure for the turbine balancing problem

Yahya Fathi and Krishna K. Ginjupalli

Department of IE and Graduate Program in OR, North Carolina State University, Raleigh, NC 27695, USA

Received June 1990; revised March 1991

Abstract: We model the turbine balancing problem as a Quadratic Assignment Problem, and develop two heuristic procedures for solving it. We then compare and contrast the overall performance of these and other procedures on an empirical basis, and recommend a general strategy for solving this problem on the manufacturing floor.

Keywords: Product design; Quadratic assignment problem; Heuristic procedures

Introduction

During the final stage of assembling an hydraulic turbine of size m, n 'blades' are placed (installed) in n fixed and equidistant 'positions' along the circumference of its 'wheel'. A schematic view of a turbine of size n is shown in Figure 1.

Although all blades have the same target weight, the natural variability in the blade manufacturing process typically causes the weight of each blade to deviate from its target value by as much as 5 to 10 percent in either direction. This, in turn, causes the center of gravity of an assembled turbine to be different from its geometric center. We refer to the distance between the center of gravity and the geometric center of an assembled turbine as its *deviation*.

Correspondence to: Y. Fathi, Department of IE and Graduate Program in OR, North Carolina State University, Raleigh, NC 27695, USA.

Naturally, for a turbine of size n and a given set of n blades, different arrangements of the blades on the wheel result in different centers of

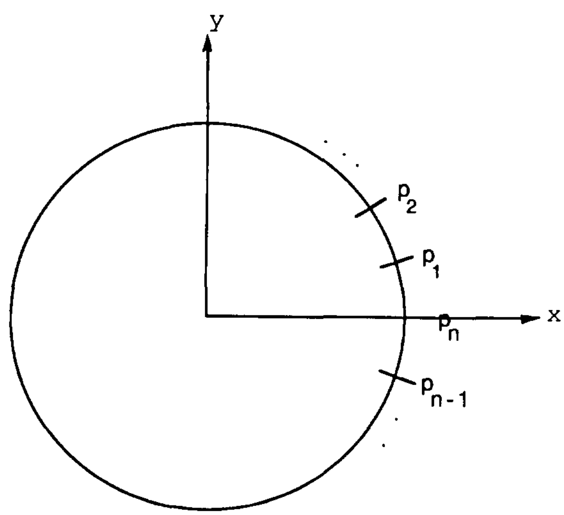

Figure 1

gravity for the turbine, and thereby different deviations. We refer to any particular arrangement of the blades as a *configuration*, and define the Turbine Balancing Problem (TBP) to be the problem of finding a configuration with smallest deviation.

This problem was first discussed by Mosevich (1986), and a heuristic procedure for solving it was proposed. This procedure basically consists of generating a relatively large number of random configurations, evaluating their respective deviations, and selecting the best among them. In short, this procedure uses the well-known Monte Carlo sampling technique to solve the problem. Computational results reported by Mosevich show that this method is a significant improvement over the manual trial and error technique previously practiced in a manufacturing firm.

Building on the work of Mosevich, Korenjak and Batagelj (1987) proposed an iterative heuristic procedure for solving the problem. This procedure starts with a randomly generated configuration which is initially referred to as the 'current configuration'. Then, at each iteration, it examines the effect of pairwise interchange of the locations of each pair of blades within the current configuration. If the effect is a net decrease in the deviation, it performs the corresponding interchange and obtains a new 'current configuration'. The procedure continues in this manner until it reaches a configuration for which no such improving pairwise interchange is possible. This configuration is labelled 'a local optimum', and the procedure terminates.

Naturally, the local optimum configuration obtained in this manner very much depends on the initial configuration used. Therefore, Korenjak and Batagelj recommend to apply this procedure several times, each time with a different (randomly generated) starting configuration, and select the best among the local optima so obtained.

Computational results reported by Korenjak and Batagelj show that this procedure is a substantial improvement over the Monte Carlo procedure of Mosevich, in terms of the quality of solutions it obtains. This result is not surprising since apparently in their experiment Korenjak and Batagelj applied the pairwise interchange heuristic to every randomly generated configuration they used in their implementation of the Monte Carlo approach (1000 configurations). The computational requirements of the pairwise interchange procedure, however, could be quite large. Korenjak and Batagelj do not report on this aspect of their experiment.

In this article, we present a mathematical formulation of the problem as a Quadratic Assignment Problem, and study its various properties. We then propose two heuristic procedures, and compare these procedures with those proposed by Mosevich (1986) and by Korenjak and Batagelj (1987), on an empirical basis.

As the test problems in our computational study we use a family of artificially constructed TBPs that we refer to as the Integer Turbine Balancing Problem (ITBP). An ITBP of size n (for $n \geq 2$) is obtained by letting the weight of the i-th blade to be equal to i for $i = 1$ to n. This is the same set of test problems used by Korenjak and Batagelj (1987). Some observations regarding the combinatorial structure of ITBP are also discussed.

Formulation as a Quadratic Assignment Problem

Consider a turbine of size n, and let w_i represent the weight of the ith blade, for $i = 1$ to n. Also let the positions on the turbine be numbered as shown in Figure 1. The angle between the radii corresponding to any two consecutive positions is $2\pi/n$ radians, and the coordinates of the j-th position (with respect to the reference axes shown in Figure 1) are $[r \cos(2\pi j/n), r \sin(2\pi j/n)]$, for $j = 1$ to n, where r is the radius of the wheel.

Let

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

These variables represent a configuration if they satisfy the following assignment constraints:

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \text{for all } j = 1 \text{ to } n,$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for all } i = 1 \text{ to } n.$$

Letting (X, Y) represent the coordinates of the center of gravity of the turbine, and for x_{ij}'s satisfying the above constraints, we have

$$WX = \sum_{j=1}^{n} r\left\{\cos(2\pi j/n) \sum_{i=1}^{n} w_i x_{ij}\right\},$$

$$WY = \sum_{j=1}^{n} r\left\{\sin(2\pi j/n) \sum_{i=1}^{n} w_i x_{ij}\right\}$$

where $W = \sum_{i=1}^{n} w_i$. Thus

$$X^2 = \left(\frac{r}{W}\right)^2 \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} w_i w_k \cos(2\pi j/n) \cos(2\pi l/n) x_{ij} x_{kl},$$

$$Y^2 = \left(\frac{r}{W}\right)^2 \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} w_i w_k \sin(2\pi j/n) \sin(2\pi l/n) x_{ij} x_{kl},$$

and

$$d^2 = X^2 + Y^2 = \left(\frac{r}{W}\right)^2 \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} w_i w_k \cos[2\pi(j-l)/n] x_{ij} x_{kl}$$

where d is the deviation corresponding to the configuration represented by the x_{ij}'s. Therefore, the TBP can be written as the following Quadratic Assignment Problem (QAP):

$$\text{minimize} \quad z = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} w_i w_k \cos[2\pi(j-l)/n] x_{ij} x_{kl}$$

$$\text{subject to} \quad \sum_{i=1}^{n} x_{ij} = 1 \quad \text{for all } j = 1 \text{ to } n,$$

$$\textstyle\sum j = 1 x_{ij} = 1 \quad \text{for all } i = 1 \text{ to } n,$$
$$n$$

$$x_{ij} = 0 \text{ or } 1 \quad \text{for all } i \text{ and } j.$$

The size of this QAP can be reduced if we arbitrarily assign one of the blades to a position (say Blade 1 to Position 1, i.e., fix $x_{11} = 1$). This restriction does not change the optimal value of the objective function, since for any configuration there exists an *equivalent* configuration in which Blade 1 is at Position 1. (We define two configurations to be *equivalent* if one can be obtained from the other by a series of 'rotation' operations. A 'rotation' operation re-assigns the blade in Position 1 to Position 2, the blade in Position 2 to Position 3, etc. This has the effect of rotating the turbine counterclockwise through an angle of $2\pi/n$ radians. Thus, a rotation operation does not change the corresponding deviation. It follows that for any configuration there exists an equivalent configuration in which Blade 1 is at Position 1. Hence by restricting our feasible region to those solutions in which $x_{11} = 1$ we indeed do not preclude any configuration.) Furthermore, fixing a blade at a position has the added advantage that the resulting (smaller) QAP does not have as many alternate optima as the original model. This is particularly important if we employ an exact algorithm to solve the QAP, since such algorithms are known to perform poorly if the problem has several alternate optima.

During the past few decades, QAP has been extensively studies by several researches, and various exact as well as inexact algorithms for this problem are published in the open literature. For surveys of these algorithms see Burkard and Stratmann (1978), Burkard (1984), and Christofides and Benavent (1989). Due to the substantial computational requirements of the exact algorithms for QAP, these techniques are recommended only for small instances of the problem ($n < 15$), while larger instances are typically solved using inexact algorithms. Burkard (1984) mentions that in some cases it may also be worthwhile to consider special cases of the QAP and develop specific algorithms to deal with these cases. The heuristic procedures discussed in the next sections are of this type.

We discuss two inexact procedures for solving the TBP, and compare these procedures with those of Mosevich (1987) and Korenjak and Batagelj (1987) on an empirical basis. We refer to these procedures as the Placement Heuristic (PH) and the Rotational heuristic (RH), respectively. PH tends to be more appropriate for smaller values of n ($n \leq 10$), while RH is more appropriate for larger values of n. Two variations of the Placement Heuristic are also presented and discussed.

Placement heuristic procedure and its variations

The Placement Heuristic (PH) is an inexact greedy procedure which is based on the QAP formulation of the TBP, and consists of exactly n steps, where n is the size of the turbine. During the first step, the heaviest blade is 'placed' arbitrarily in a position on the turbine. During each of the subsequent steps the next heaviest blade is permanently 'placed' in a position which is determined on the basis of the relative contribution of the 'placement' on the objective function of the QAP with respect to the blades already placed. Details of this procedure are discussed below. Throughout the remainder of this article, without loss of generality, we assume that $w_1 \geq w_2 \geq \cdots \geq w_n$.

Step 1 (Initialization). Place Blade 1 in Position 1. Let $P = \{2, 3, \ldots, n\}$ be the current set of available (unoccupied) positions. Go to Step 2.

general Step k (for $k = 2$ to n). Place Blade k in Position j^* such that

$$\sum_{i=1}^{k-1} w_i \cos\{2\pi[j^* - l(i)]/n\} = \underset{j \in P}{\text{minimum}} \sum_{i=1}^{k-1} w_i \cos\{2\pi[j - l(i)]/n\}$$

where $l(i)$ = Position of Blade i, for $i = 1$ to $k - 1$. Let $P = P \backslash j^*$. If $k = n$, stop; otherwise go to Step $k + 1$.

The computational requirement of the Placement Heuristic is of order n^3.

Variations of this procedure are obtained by further exploring the consequences of placing each blade in each unoccupied position, in terms of its impact on the objective function, prior to permanent placement of that blade. Such an exploration typically requires more computational effort, but could result in better overall configurations. We discuss two such variations, and refer to them as Level 1 Placement Heuristic (PH1) and Level 2 Placement Heuristic (PH2), respectively. The first step of either of these procedures is identical to that of the PH discussed above. Thus we only discuss the general Step k for each of these procedures.

General Step k of PH1 *(for $k = 2$ to n).* This step consists of $n - k + 1$ substeps. During the i-th substep, for $i = 1$ to $n - k + 1$, temporarily place Blade k in the i-th unoccupied position (i-th element of the set P), and place the remaining $n - k$ blades according to the PH procedure. This results in a configuration which we refer to as the i-th temporary configuration. Compute the corresponding deviation and associate it with the i-th unoccupied position. We then choose the unoccupied position $j^* \in P$ with the smallest associated deviation, and permanently place Blade k in that position. Let $P = P \backslash j^*$. If $n = k$, stop; otherwise go to Step $k + 1$.

General Step k for PH2 *(for $k = 2$ to n).* This step is similar to Step k of PH1, except that the temporary configuration in each substep is obtained using PH1 rather than PH.

It can be verified that the computational requirements of PH1 and PH2 are of order n^5 and n^7, respectively.

In general, the Level l Placement Heuristic (PHl) is obtained by using the Level $l - 1$ Placement Heuristic during the substeps. It is only natural to expect the quality of the configuration produced by PHl to improve as l grows larger (in fact at $l = n - 1$ this procedure produces an optimal configuration), but it should be noted that the corresponding computational requirements increase very rapidly. In our computational experiments we observed a substantial increase in the computer ellapsed time when we increased l from 2 to 3. Based on these experiments we do not recommend to use values of l larger than 2.

Table 1 contains the deviations obtained by applying PH, PH1, and PH2 on the test problems (ITBPs of size 5 through 24). All programs for these procedures (as well as those for other procedures which are discussed later) were written by the authors in BASIC, and ran on a VAX 11/750 computer in the Department of Industrial Engineering at North Carolina State University. The computer ellapsed time for PH2 (the more ellaborate of the three procedures) and for the largest ITBP solved ($n = 24$) was less than 2 seconds.

In Table 1 we also include the best deviations obtained by the algorithm of Karenjak and Batagelj (1987), for ITBPs of size 5 through 20 as reported in their article, to allow a direct comparison. These results are based on 1000 applications of the pairwise switching algorithm for each value of n. It should be noted that the computational requirement of this algorithm is substantially higher than that of PH2.

It can be observed that the solutions obtained by PH2 compare well with those obtained by the algorithm of Korenjak and Btagelj for smaller values of n, but they start to deteriorate as n grows larger. In light of the fact that PH2 is greedy in nature, this observation is not surprising. In the next section we present another inexact procedure which we refer to as the Rotational Heuristic (RH). RH is particularly appropriate for larger values of n.

Rotational heuristic procedure

This procedure is based on partitioning the blades into subsets of equal cardinality. Thus, it is applicable only if the total number of blades is not a prime number. Details of this procedure vary slightly depending on the number of subsets involved. We refer to the procedure with k subsets ($k \geq 2$) as the k-Rotation Heuristic, or simply RHk. We start our discussion in this section with a detailed statement of RH2. A general statement of RHk for $k > 2$ follows. Throughout this section, as in the previous section, we assume that $w_1 \geq w_2 \geq \cdots \geq w_n$.

Naturally, RH2 is applicable only if n is divisible by 2. It consists of three steps as discussed below.

Step 1. Divide the baldes into two subsets S_1 and S_2, with an equal number of blades in each subset. Division of the blades should be done in such a manner that the weights of blades in the two subsets have similar profiles. In our experiments we let

$$S_1 = \{w_1, w_3, \ldots, w_{n-1}\} \quad \text{and}$$

$$S_2 = \{w_2, w_4, \ldots, w_n\}.$$

Other divisions of the blades can also be used in this step. For the test problems under consideration in our experiment this particular division resulted in a good overall performance of the algorithm.

Step 2. For each of the subsets S_1 and S_2 find a good configuration using the PH2 procedure (we refer to each configuration so obtained as a subconfiguration). Then combine the resulting two subconfigurations in the following manner:

(i) Place the first subset of blades (S_1) according to the subconfiguration obtained in Step 1, starting in Position 1 and using alternate positions in a clockwise direction (i.e., using Position 1, 3, 5, ..., $n-1$ in order).

(ii) Place the second subset of baldes (S_2) according to its subconfiguration in the remaining positions, starting in Position 2 and moving once in a clockwise direction and once in a counterclockwise direction (i.e., once using Positions 2, 4, ..., n, in order, and once using Positions 2, n, $n-2, \ldots, 4$ in order). This results in two different configurations for the turbine. We refer to these configurations as the 'initial configurations'.

Step 3. Try to improve each of the initial configurations obtained in Step 2 by rotating the subset S_1 one position at a time in a clockwise direction, while keeping S_2 fixed. By rotation S_1 one position in a clockwise direction we mean that each blade in this subset is assigned to the position of its adjacent blade within the subset in a clockwise direction (i.e., the blade in Position 1 moves to Position 3, the blade in Position 3 moves to Position 5, etc.). This one-step-at-a-time rotation results in $\frac{1}{2}n$ distinct configurations for each initial configuration (a total of n distinct configurations). For each configuration obtained in this manner compute the corresponding deviation, and select the configuration with the smallest deviation.

The motivation for rotating a subset (Step 3) is to obtain a configuration in which the centers of gravity for the two subsets S_1 and S_2 are (almost) alligned with the center of the turbine and positioned on its opposite sides. This, in turn, results in the overall center of gravity for the entire turbine to be much closer to its geometric center. The computational effort required for this rotation is reasonably small, as we need only to compute the deviation for n distinct configurations.

RHk (for $k > 2$) is quite similar to RH2, except that initially we partition the blades into k subsets S_1 through S_k (where n must be divisible by k). For each subset S_i, for $i = 1$ to k, we obtain a good subconfiguration by using either an exact algorithm or a heuristic procedure such as those discussed earlier (we used PH2 in our experiment). We then combine these subconfigurations in a manner similar to Step 2 in RH2, to obtain various 'initial configurations' (i.e., place the first subset of blades (S_1) according to its subconfiguration in Positions 1, $1+k$, $1+2k, \ldots, 1+(n/k-1)k$; then place the ith subset of blades (S_i, for $i=2$ to k) according to its subconfiguration, starting in Position j (once for each $j=2$ to k) and using every kth position, once in a clockwise direction and once in a counterclockwise direction). For each initial configuration so obtained, 'rotate' $k-1$ of these subsets while keeping one subset fixed (in a manner similar to Step 3 in RH2). This results in a number of distinct configurations from which we select the one with smallest deviation.

The computational requirements of this method could be quite large for larger values of k and n. There are a total of $2^{k-1}(k-1)!$ distinct initial configurations, with $(n/k)^{k-1}$ different configurations resulting from rotating each initial configuration. Hence this procedure obtains $(2n/k)^{k-1}(k-1)!$ distinct configurations. For each configuration the value of the corresponding deviation must be computed.

We used RHk (for $k=2$, 3, 4) to solve the same set of problems as we solved earlier using the Placement Heuristic procedures. For each value of n we used every possible value of k, and the results are presented in Table 1 alongside the previous results. It can be observed that in most instances the deviations obtained by this procedure are at least as good as those obtained by other procedures, and they tend to get strictly better for larger values of n. The computational requirements of this procedure were also small (elapsed time less than 2 seconds).

Remarks on the test problem ITBP

From the compuational results presented in Table 1 it is clear that for some values of n the optimal deviation for the ITBP of size n is in fact equal to zero (center of gravity and geometric center of the turbine coincide). We refer to a configuration with zero deviation as a *perfect configuration*. It can also be observed in Table 1 that the rotational heuristic procedure finds perfect configurations for some values of n for which other procedures fail to do so. However, there still remain some values of n for which even the rotational heuristic does not find perfect configurations. The following lemma states a set of suffi-

Table 1
Deviations obtained by various heuristic procedures for ITBPs of size 5 through 24, with wheel radius $r=1000$

n	Korenjak and Batagelj [a]	Placement Heuristics			Rotational Heuristics		
		PH	PH1	PH2	RH2	RH3	RH4
5	29.93	29.93	29.93	29.93			
6	0.00	0.00	0.00	0.00	0.00	0.00	
7	2.73	2.73	2.73	2.73			
8	3.64	30.06	3.64	3.64	30.06		30.6
9	0.63	13.94	0.63	0.63		20.48	
10	0.00	0.00	0.00	0.00	0.00		
11	0.02	2.09	2.09	0.89			
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.04	8.00	2.74	1.79			
14	0.00	0.00	0.00	0.00	0.00		
15	0.26	15.68	0.84	0.38		0.00	
16	0.02	7.49	0.54	0.23	0.37		3.10
17	0.05	5.03	0.14	0.11			
18	0.03	4.06	0.99	0.92	0.00	0.00	
19	0.05	0.73	0.72	0.40			
20	0.04	1.94	0.30	0.00	0.00		0.0
21	n/av	3.05	0.82	0.75		0.00	
22	n/av	2.95	0.74	0.56	0.00		
23	n/av	1.63	0.31	0.13			
24	n/av	2.46	0.87	0.49	0.00	0.00	0.00

[a] n/av: not available.

cient conditions for the existence of a perfect configuration for the ITBP, thus providing a partial explanation for this phenomenon.

Lemma 1. *An* ITBP *of size n has a perfect configuration if n is divisible by* 2 *but not divisible by* 4, *or if n is divisible by* 3 *but not divisible by* 9.

Proof. See Ginjupalli (1989).

We further conjecture that an ITBP of size n has a perfect configuration if n is divisible by k but not divisible by k^2, for any prime number $k \geq 2$. The values of n which are considered in our experiment and satisfy this condition are 6, 10, 12, 14, 15, 18, 20, 21, 22, 24. It is interesting to note that RHk obtains an optimal solution (perfect configuration) for every one of these instances of the problem, while it performs relatively poorly in all other instances where it is applicable but where n does not satisfy this condition.

Concluding remarks

Based on our limited computational experiment we observe that the rotational heuristic procedure seems to outperform the other procedures in terms of the quality of the configurations it obtains (with a few exceptions as discussed earlier). This observation, in conjunction with the fact that the computational requirements of RHk for small values of k are relatively small, makes this procedure an appropriate choice for solving the Turbine Balancing Problem on the manufacturing floor, if it is applicable (i.e., if n is not a prime number) and particularly for large values of n. For smaller values of n, and for cases where RHk is not applicable, we recommend to use either PH2 or the pairwise switching algorithm of Korenjak and Batagelj.

Acknowledgement

We would like to thank Dr. S.E. Elmaghraby for his support and many helpful suggestions during the early stages of this research. This research was partially supported by ARO Contract No. DAAL03-86-K-0039.

References

Burkard, R. (1984), "Quadratic assignment problems", *European Journal of Operational Research* 15, 283–289.

Burkard, R., and Stratmann, K. (1978), "Numerical investigations on Quadratic Assignment Problems", *Naval Research Logistics Quarterly* 25, 129–148.

Christofides, N., and Benavent, E. (1989), "An exact algorithm for the quadratic assignment problem on a tree", *Journal of Operations Research* 37, 760–768.

Ginjupalli, K. 91989), "Balancing a Hydraulic Turbine", Master Thesis, Department of Industrial Engineering, North Carolina State University, Raleigh, NC.

Korenjak, S., and Batagelj, V. (1987), "Turbine balancing problem", Technical Report, Department of Mathematics, University of Edvard Kardelj of Eljubljana, Yugoslavia.

Mosevich, J. (1986), "Balancing hydraulic turbine runners – A discrete combinatorial optimization problem", *European Journal of Operational Research* 26, 202–204.