International Workshop On Combinatorial Algorithms
IWOCA

http://www.iwoca.org—Problems section.

Balanced Mobiles

Yassine Hamoudi
ENS Lyon

Sophie Laplante
IRIF, Université Paris Diderot

Roberto Mantaci
IRIF, Université Paris Diderot

presented at IWOCA 2016
18 Aug. 2016

1 The problem

A mobile is a full binary tree in which each leaf has a (positive) weight. Actual realizations of these objects can be found in modern art (Calder) or above toddler beds.

For each internal node u of a mobile, define the imbalance of u as the difference (in absolute value) between the sum of all weights of the leaves of the left subtree of u and the sum of all weights of the leaves of the right subtree of u. Define the imbalance $\Delta(M)$ of a mobile M as the sum of the imbalances of all its nodes. If $\Delta(M) = 0$, the mobile is perfectly balanced.

The problem BalancedMobiles consists in constructing, for a given (multi-)set of weights $p_1, p_2, \ldots, p_n$, a mobile M whose leaves have weights $p_1, p_2, \ldots, p_n$ (in any possible order) and whose imbalance $\Delta(M)$ is as small as possible.

2 Previous work

The variant of the problem in which the order of the weights is fixed has the same structure as the well-known problem of determining the best possible way to multiply a sequence of matrices. Both problems can be solved with a dynamic programming algorithm in polynomial time ([God73]). Indeed, noting by $C[i,j]$ the minimum possible cost to compute the matrix product $A_i \times A_{i+1} \times \ldots \times A_j$ (where A_t is of size $d_t \times d_{t+1}$) and by $D[i,j]$ the imbalance of the optimal mobile with weights $p_i, p_{i+1}, \ldots, p_j$, one has the two following formulas :

$$C[i,j] = \min_{i<k<j} \{C[i,k] + C[k,j] + d_i d_{k+1} d_{j+1}\}$$

$$D[i,j] = \min_{i<k<j} \left\{ D[i,k] + D[k,j] + \left| \sum_{t=i}^{k} p_t - \sum_{t=k+1}^{j} p_t \right| \right\}$$

The algorithm for the mobiles is obviously slower than the algorithm for the matrices by a linear factor.

BalancedMobiles also bears resemblance to the optimization version of the Partition-Problem (NP-hard), as well as to the well-known Huffman algorithm [Huf52] that constructs in polynomial time the prefix code having the shortest average length for a given set of symbol frequencies $p_1, p_2, \ldots, p_n$.

This suggests two attempts to solve the BalancedMobiles problem. The first is a top-down approach that applies the PartitionProblem recursively at each node of the tree, starting from the root. We call this the Partition algorithm. The second one is a bottom-up approach *la Huffman* that builds the tree by joining the two subtrees having the smallest weights. We call this the Smallest algorithm.

3 Our results

At the current state of our research [HLM15], we know that the Smallest algorithm constructs the optimal mobile in polynomial time for particular classes of mobiles, namely the mobiles with $\Delta(M) = 0$, those with $\Delta(M) = 1$, those whose leaves have weights equal to powers of 2, those whose leaves have all equal weights. For the latter case, the Partition algorithm also constructs an optimal solution in polynomial time.

However, neither of these approaches always constructs an optimal solution and we can prove that Smallest does not construct an approximate solution. We also have given two algorithms solving the problem in all cases in exponential time (one is a relaxation of Smallest, and the other one is based on integer linear programming), but we do not know if BalancedMobiles is NP-hard.

We also studied the quantity Δ_n, which is the optimal imbalance when the weights $p_1, p_2, \ldots, p_n$ are all equal to 1. We proved that the following recurrences characterize Δ_n.

$$\begin{cases} \Delta_1 = 0 \\ \Delta_{2n} = 2\Delta_n \\ \Delta_{2n+1} = 1 + \Delta_n + \Delta_{n+1} \end{cases} \quad \text{and} \quad \begin{cases} \Delta_1 = 0 \\ \Delta_{n+1} = \Delta_n + |n|_0 - |n|_1 + 1 \end{cases}$$

where $|n|_0$ (resp. $|n|_1$) is the number of 0 (resp. 1) is the binary representation of n. We also obtained a closed-form expression:

$$\Delta_n = 2 \cdot (n \mod 2^k) + \sum_{i=0}^{k-1} (-1)^{b_i} \cdot (n \mod 2^{i+1})$$

where $b_k b_{k-1} \ldots b_0$ is the binary representation of n. The Δ_n function also has interesting symmetry properties.

Open Questions

1. Is BalancedMobiles NP-hard?
2. Does the Smallest algorithm work for other classes of mobiles?

3. Is it possible to obtain concise formulas for quantities other than Δ_n (for instance, when the weights are powers of 2)?

We also do not know anything about the variant of the problem where the shape of the mobile is fixed and one has just to find the permutation of the weights that minimizes the imbalance.

References

[God73] Sadashiva S. Godbole. On efficient computation of matrix chain products. *IEEE Trans. Comput.*, 22(9):864–866, September 1973.

[HLM15] Yassine Hamoudi, Sophie Laplante, and Roberto Mantaci. quilibrage dun arbre pondr : le problme Balanced Mobile. Technical report, ENS Lyon – IRIF, Universit Paris Diderot, August 2015. `http://perso.ens-lyon.fr/yassine.hamoudi/wp-content/uploads/2015/08/BalancedMobile_Rapport.pdf`.

[Huf52] D.A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, Sept 1952.

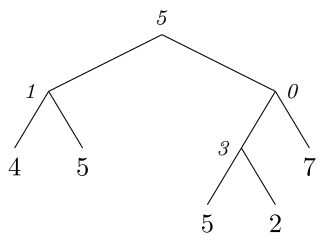

Figure 1: A mobile of imbalance $\Delta(M) = 1 + 5 + 3 + 0 = 9$ whose leaves have weights 4, 5, 5, 2 and 7.

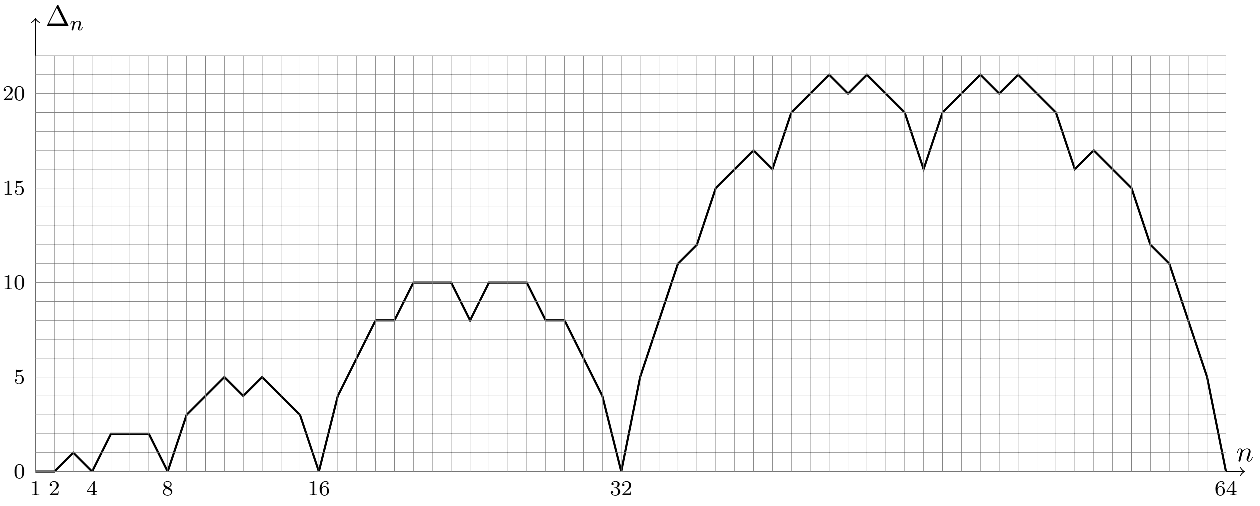

Figure 2: The Δ_n imbalance.