DIFFERENTIAL BALANCED TREES AND (0,1) MATRICES[1]

H. Sahakyan, L. Aslanyan

Abstract: Links and similarities between the combinatorial optimization problems and the hierarchical search algorithms are discussed. One is the combinatorial greedy algorithm of step-by-step construction of the column-constraint (0,1) matrices with the different rows. The second is the base search construction of databases, - the class of the well known weight-balanced binary trees. Noted, that in some approximation each of the above problems might be interpreted in terms of the second problem. The constraints in matrices imply the novel concept of a differential balance in hierarchical trees. The obtained results extend the knowledge for balanced trees and prove that the known greedy algorithm for matrices is applicable in the world of balanced trees providing optimization on trees in layers.

Keywords: search, balanced trees, (0,1)-matrices, greedy algorithm.

1. Introduction

In this paper a new class of weight-balanced trees [K, 1973] is introduced and investigated. In some sense these are extensions of the concept of the bounded balanced trees. Bounded balanced trees were analysed in various publications, e.g. [R, 1977], being the main data structure of search in dynamic databases. In Section 2 below the height estimate for bounded-balanced trees is considered and an estimate for the weight-balanced trees with the newly introduced differential balances and constraints is obtained.

The theory of weight-balanced trees is very rich. Practically this is also the base model of hierarchical search and decision support. In search, several restrictions in terms of balances are applied in a dynamic environment with insertion of new and deletion of obsolete search elements. The balances in nodes are under the change during this process. In a dis-balanced node rotations are used to correct the situation. Several queries, related to these models are traditional. Which is the tree height in a given balance and in a given set of search elements? Which is the average path length in a search tree? A particular new postulation is the following. Is it possible to construct a tree or to construct all the trees that may appear in a search model with the given constraints? This is a particular interest of the current paper.

The stated problem will be studied in several extensions, which are also a typical element of search models. E.g. - some specific classes of balanced trees, called trees of bounded heights, introduced in [A, 1989], [A, 1999].

The concept of bounded-balance is extended in Section 3, defining layer-constraint balanced trees. The idea of layer-constraints is then developed in Section 4, considering a practical extension of the concept of weight-balanced trees - defining summary balances for tree layers. This structure is related to mentioned combinatorial problem – constructing the constraint based (0,1)-matrices with different rows. In [S, 1986], [S, 1995] a greedy algorithm is constructed for solving the mentioned combinatorial problem and it is proven optimal in local steps. The algorithm for solving this problem is reducible to the constructing of weight-balanced trees by the given summary differential balances in layers. Similarly, in the world of balanced trees this proves a heuristic optimization on trees in layers.

2. Bounded-balanced Trees

Let T_m be a non-empty extended binary tree [R, 1977] with m leaves, and T_l and T_r are the left and right root-subtrees of T_m. We denote by l and r the numbers of leaves of T_l and T_r (called weights) and assume that $l > 0$ and $r > 0$. Then $m = l + r$.

Definition [R, 1977]. The fraction l / m is called the balance (left, fractional) of T_m in root vertex, being denoted by $\beta(T_m)$. $\beta(T_m)$ expresses the ratio weight of the left root-subtree and it obeys the condition $0 < \beta(T_m) < 1$.

[1] The research was supported by INTAS 00-397 and 00-626 Projects.

Definition [R, 1977]. For a given α, $0 \le \alpha \le 1/2$, T_m is called an α-balanced tree (or a tree from $WB[\alpha]$) if

1) $\alpha \le \beta(T_m) \le 1-\alpha$,
2) left and right subtrees of T_m belong to $WB[\alpha]$.

We assume by definition that the empty binary tree belongs to $WB[\alpha]$.

$WB[0]$ is the set of all binary trees and $WB[1/2]$ is the set of all perfectly balanced (with the equal left and right subtrees weights in each node) binary trees, and this is possible, when the number of leaves has the form 2^k $(k \ge 1)$.

The maximum possible height $h_\alpha(m)$ of trees from $WB[\alpha]$ is estimated in [R, 1977] – by the consideration of the most asymmetric trees of $WB[\alpha]$:

$$h_\alpha(m) \le \frac{\log m}{\log(1/(1-\alpha))}$$

(1)

It is also important to treat the question: given the binary trees with m leaves and with heights, restricted by a given number n, then - how “unbalanced” may be the trees, - which is the allowable minimum value for α? The answer (in a form of a sufficiency condition) is given by the lemma below, using the monotony of (1).

Lemma 1. If $\alpha \ge 1-\frac{1}{\sqrt[n]{m}}$, then $h_\alpha(m) \le n$.

The $\alpha \ge 1-\frac{1}{\sqrt[n]{m}}$ implies $\frac{1}{1-\alpha} \ge 2^{\frac{\log m}{n}}$, and then $\log(1/(1-\alpha)) \ge \frac{\log m}{n}$, and $\frac{\log m}{\log(1/(1-\alpha))} \le n$. For those α, $h_\alpha(m) \le \frac{\log m}{\log(1/(1-\alpha))} \le n$ by (1).

Now let us turn to the concept of balances in terms of differences of weights between subtrees.

Definition. The difference $r-l$ is called the differential balance (right) of T_m in the root vertex, denoted by $\delta(T_m)$. It obeys the following condition: $1-m < \delta(T_m) < m-1$.

Definition. For a given d, $0 \le d < m-1$, T_m is called differential-balanced tree with balance d (or a tree from $WDB[d]$) if

1) $-d \le \delta(T_m) \le d$,
2) left and right subtrees of T_m belong to $WDB[d]$.

The two balance schemes are tightly related. Let us formulate the base relations between the fractional and differential balances.

Let T_m be an extended binary tree with m leaves, and v_i is a vertex (not leaf) of T_m. We denote by m_i the weight of subtree rooted at v_i, and by l_i and r_i - the weights of its left and right subtrees, correspondingly. Starting at this point we will assume also, that l_i-s are not greater than r_i-s.

If T_m is an α -balanced tree then for each vertex v_i we have $\frac{l_i}{m_i} \geq \alpha$ by the definition. Let's estimate the weight differences between right and left subtrees for each v_i:

$r_i - l_i = m_i - 2l_i = m_i(1 - 2\frac{l_i}{m_i}) \leq m_i(1 - 2\alpha)$. Hence

$r_i - l_i \leq \max_i m_i(1 - 2\alpha) = m(1 - 2\alpha)$.

Conclusion is that an α -balanced tree T_m is a differential balanced tree with $d = m(1 - 2\alpha)$.

Now let T_m is a differential balanced tree with balance d. For each vertex v_i, $r_i - l_i \leq d$ by the definition. Let's estimate the fraction $\frac{l_i}{m_i}$;

$\frac{l_i}{m_i} = \frac{m_i - (r_i - l_i)}{2m_i} = \frac{1}{2}(1 - \frac{r_i - l_i}{m_i}) \geq \frac{1}{2}(1 - \frac{d}{m_i}) \geq \frac{1}{2}(1 - \frac{d}{m})$.

Thus, a differential balanced tree T_m with balance d, is an α -balanced tree, with $\alpha = \frac{1}{2}(1 - \frac{d}{m})$. A more correct estimate is:

$$\alpha = \frac{1}{2}(1 - \max_i \frac{\min\{d, m_i - 2\}}{m_i}).$$

Next we consider the estimate of height of trees from $WDB[d]$, constructing the most asymmetric trees in this class. On each layer the subtrees with greatest weights have been partitioned into the subtrees with maximization of weights differences. Then the height estimate is the length of these "maximum weighted" branches.

On the first layer we get subtrees of weights $\frac{m+d}{2}$ and $\frac{m-d}{2}$. We will follow only the branch of weight $\frac{m+d}{2}$. On the next layer we will get a subtree of weight $\frac{\frac{m+d}{2} + d}{2} = \frac{m+3d}{4}$. In continuation, let k is the minimal index, where the maximal subtree weight becomes less than d. At that point the maximum weight doesn't exceed $\frac{m + (2^k - 1)d}{2^k}$.

$$\frac{m + (2^k - 1)d}{2^k} = \frac{m-d}{2^k} + d \text{, therefore } \frac{m-d}{2^k} < 1 \text{, and } k > \log(m - d).$$

Resuming, we receive, that after at most $\log(m - d) + 1$ steps the weight of maximal subtree is less than d. If $d \leq 1$, then the tree construction is complete, and we get a tree with the height estimate $\log(m - d) + 1$. Otherwise we continue the process, with the arbitrary partition of subtrees. At most $d - 1$ steps will be required. We receive the following final estimation – the heights of trees from $WDB[d]$ are restricted by $\log(m - d) + d$.

Now we treat the question about the constraints on balances when given that the heights are restricted. Let us consider the binary trees with m leaves, and heights, restricted by the given number n. The counterpart of Lemma 1 is the following proposition:

Lemma 2. If the differential balance d obeys: $\log(m-d)+d \le n$, then the height of tree with m leaves is restricted by n.

A practical note. The concept of differential balancing is reasonable to apply on trees as far as the weights of subtrees are greater than d , therefore - on layers of at most $\log(m-d)+1$ far from the root.

3. Layer-constrained Weight-balanced Trees

At this point the concept of differential balances is introduced and the general comparison with the base scheme – the weight-balanced trees is outlined. The particular properties of differential balances are that these are flexible on tree layers. The balance constraints may vary from layer to layer and/or the constraints might be given in terms of summary balances. In some cases it is important to apply these structures in the traditional case of the weight-balanced trees. These issues are considered below.

Definition. For a given α_i, $0 \le \alpha_i \le 1/2$, we say that T_m is α_i-balanced on layer i, if for each subtree T_{i_j} - rooted at layer i, $\alpha_i \le \beta(T_{i_j}) \le 1-\alpha_i$.

Definition. Given numbers $\alpha_0,\cdots,\alpha_k$, where $0 \le \alpha_i \le 1/2$, $i=0,\cdots,k$. We say that T_m is a tree from class $WB[\alpha_0,\cdots,\alpha_k]$, if T_m is α_i-balanced on layer i.

The leaves may be composite in $WB[\alpha_0,\cdots,\alpha_k]$ (when k -sequences are not enough to differentiate the nodes, the composite nodes may remain consisting of sets of virtual leaves). On the other hand, part of the balance values (a last portion) may be redundant. Consideration of the most asymmetric trees and paths in $WB[\alpha_0,\cdots,\alpha_k]$ gives the following estimation: the weights of subtrees (virtual at this point) of k -th layer are restricted in size by $(1-\alpha_0)(1-\alpha_1)\cdots(1-\alpha_k)m$. If there exists h, $h \le k$, such that $(1-\alpha_0)(1-\alpha_1)\cdots(1-\alpha_h)m \le 2$, then the height of the tree is restricted by h.

Now we consider layer constrained weight-balanced trees in sense of differential balances.

Definition. For a given d_i, $0 \le d_i < m-1$, we say that T_m has d_i differential balance on layer i, if for each subtree T_{i_j} - rooted at layer i, $-d_i \le \delta(T_{i_j}) \le d_i$.

Definition. Given numbers $d_0,\cdots,d_k$, where $0 \le d_i < m-1$, $i=0,\cdots,k$. We say that T_m is a tree from class $WDB[d_0,\cdots,d_k]$, if T_m has differential balances d_i on layers i.

Similarly with the class $WB[\alpha_0,\cdots,\alpha_k]$, the leaves may be composite, or some last balance values may be redundant for trees of $WDB[d_0,\cdots,d_k]$. Consider the most asymmetric trees of the class $WDB[d_0,\cdots,d_k]$. Using reasoning, similar to the used above, we get that the weights of subtrees on the k -th layer of trees are restricted by $\dfrac{m+d_0+2d_1+\cdots+2^k d_k}{2^{k+1}}$.

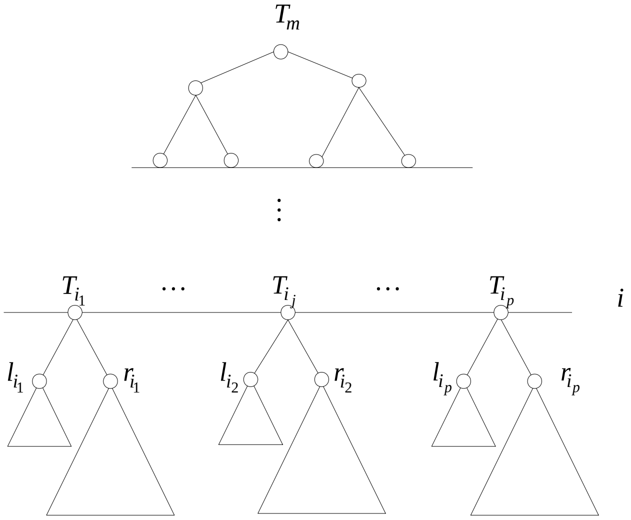
If there exists h, $h \le k$, such that $\frac{m + d_0 + 2d_1 + \cdots + 2^h d_h}{2^{h+1}} \le 1$, then the overall height of tree is restricted by h. It is easy to see that $d_0, \cdots, d_k$ must obey in this case very specific restrictions, which limits the selection and the meaning of differential balances.

4. Summary Differential Balanced Tress and (0,1)-matrices

Definition. Given numbers $D_0, \cdots, D_n$ where $0 \le D_i < m-1,\ i = 0, \cdots, n$. We say that T_m has D_i summary differential balance on i -th layer, if $R_i - L_i \le D_i$, where R_i is the sum of weights of the all right subtrees rooted at the layer i and L_i is the same sum for the left subtrees.

Definition. T_m is called $\{D_0, \cdots, D_n\}$ summary differential balanced tree if the summary balance on i -th layer equals to D_i, $i = 0, \cdots, n$.

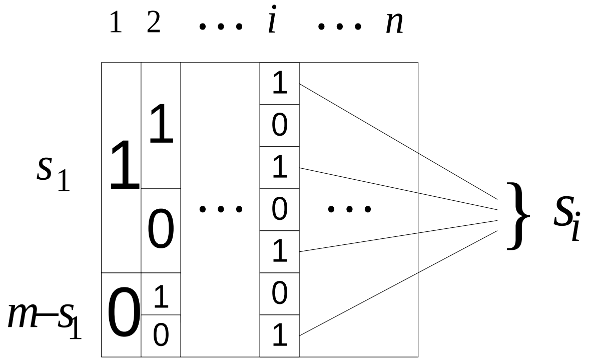
Let $T_{i_1}, \cdots, T_{i_p}$ are subtrees rooted at i -th layer having the weights $m_{i_1}, \cdots, m_{i_p}$ correspondingly, and let l_{i_j} and r_{i_j} are the weights of left and right subtrees of T_{i_j}. Then $R_i = \sum_{j=1}^{p} r_{i_j}$ and $L_i = \sum_{j=1}^{p} l_{i_j}$.



Usually the balance criterion restricts the weights of subtrees, making it possible to optimize the height of the tree. The summary balance allows any weights for subtrees, requiring only satisfying the given summary constraints on layers. This is a week form constraints for constructing an optimized decision tree. The most asymmetric trees are very diverse in this case. In next point the combinatorial origin of these differential schemes will be described. In classes of summary balanced trees the problem is in existence of a binary tree with the given characteristics $D_0, \cdots, D_n$, and in case of existence – the algorithmic construction of such trees. In special cases the issue of construction of trees might be the interest, when an additional functional for optimization is given. In particular, optimality might be required as for subtree weights on layers, for number of subtrees on layers, a special functional optimization, etc.

Here is the combinatorial counterpart of the scheme of the summary differential balances. Given an integer vector $S = (s_1, \cdots, s_n)$, where $0 \le s_i \le m$, $i = 1, \cdots, n$. The interest is in (0,1) -matrices of size $m \times n$ (m is the number of rows) with s_i 1's in i -th column and with different rows. This is the existence problem.

The corresponding optimization problem is in minimization of the number of the possible repeated rows. The problem might be solved, in particular, by algorithms, constructing the matrices in a column-by-column fashion, by partitioning the sets of similar (equal) rows received in a previous step. The first column has been constructed substituting s_1 1's and $m-s_1$ 0's. Without loss of generality we assume that the 1's are substituted on the first s_1 rows. The second column has been constructed by partitioning the intervals (sets of similar rows) of the first column (of lengths s_1 and $m-s_1$) - substituting 1's and 0's on these intervals such that the summary length of one-intervals (where all 1's are substituted), is equal to s_2, and the summary length of zero-intervals (where 0's are substituted), is equal to $m-s_2$. The partitioning of intervals for current k -th column is arbitrary, providing only that the summary length of all one-intervals is equal to s_k. Such construction provides the following property: for each pair of rows, (i, j), where rows i and j belong to different intervals, we have that the i -th and j -th rows are different. Within each interval we have sets of equal rows. The intervals with 1 length in each column don't participate in further partitioning, but they are used (substituting 1 or 0) to provide the summary values s_k and $m-s_k$ on the current k -th column. When in some column there are all 1 length intervals, then all rows are different, and the required matrix is constructed. The remainder columns might be constructed arbitrarily. The graphical scheme is the following:



This is the existence problem as was mentioned. In the similar optimization problem the row repetitions is to be minimized. Let, in a current state of construction we have intervals of lenghts $m_{n_1},\cdots,m_{n_p}$ (greater than 1) on the n-th column. Each of the $m_{n_1},\cdots,m_{n_p}$ intervals consists of the same rows repetitions. The number of pairs of rows - (i, j), where rows i and j are the same, equals $\sum_{j=1}^{p} C^2_{m_{n_j}} = \frac{1}{2}\sum_{j=1}^{p} m_{n_j}(m_{n_j}-1)$, so this is the subject for optimization.

The construction of (0,1)-matrices might be represented by binary trees. We construct a tree T_m with m leaves. The matrix with m rows corresponds to the root vertex. The submatrix with the first s_1 rows from the first step corresponds to the right subtree, and the submatrix with $m-s_1$ rows corresponds to the left subtree, etc. When the current submatrix consists of a single row, we get a leaf. When for any k, $k \le n$, the k -th layer contains leaves only, then the construction is completed. Otherwise as a result of construction on n-th layer we receive a set of subtrees of weights $m_{n_1},\cdots,m_{n_p}$. The constructed trees belong to the class of summary differential balanced trees with summary balances $D_1,\cdots,D_n$, for the given balances $D_i = s_i-(m-s_i), \quad i=1,\cdots,n$.

[S, 1986], [S, 1995] provide an approximation greedy algorithm, which constructs the target (0,1)-matrices in the above described column-by-column fashion of partitioning. The algorithm provides the optimal construction of each column – i.e. the construction, which provides the maximal number of new (i, j) pairs of different rows in each step. It is proven that the optimal construction of each column is provided by partitioning, which distributes the difference $s_k-(m-s_k)$ "homogeneously" on all current non atomic intervals. Returning to the trees terminology, the matrix constructed by the greedy algorithm implies subtrees on each layer of tree, partitioned such that the difference $R_i-L_i=D_i$ is distributed "equally" on all current subtrees.

So this describes the construction of trees in class of summary differential balanced trees providing the local optimum for the functional from the related combinatorial problem of (0,1)-matrices.

A last note. Let the subtrees of i-th layer with weights $m_{i_1},\cdots,m_{i_p}$ are partitioned into the subtrees with weights $l_{i_1},r_{i_1},\cdots,l_{i_p},r_{i_p}$ correspondingly. We denote $d_{i_1}=r_{i_1}-l_{i_1},\cdots,d_{i_p}=r_{i_p}-l_{i_p}$. Then the differential balance on i-th layer is equal to $\max\limits_{1\le j\le p} d_{i_j}$. Since D_i is distributed "equally" by the greedy partition, $\max\limits_{1\le j\le p} d_{i_j}$ will have the minimum value among all possible partitions. This is the following property: an algorithm, which is locally optimal by means of (0,1)-matrices, is locally optimal also by means of construction of trees of minimal height in class of summary differential balanced trees.

Conclusion

Resuming, - in problem of constructing the summary balanced binary trees with given differential balances of layers, and with height minimization, it is possible to apply the given above combinatorial greedy algorithm, and then the resulting tree has a property that the maximal value of the differential balances on tree layers are optimal - minimal. In terms of search trees this is an extension of perfect balanced trees on layers, when additional constraints are applied.

Bibliography

[K, 1973] D. Knut,.The Art of Computer Programming, vol.3. Sorting and Searching, Addison-Wesley Publishing Company, 1973.

[R, 1977] E. Reingold, J. Nivergelt and N. Deo. Combinatorial Algorithms, Theory and Practice, Prentice-Hall, 1977.

[A, 1989] A. Andersson. Improving Partial Rebuilding by Using Simple Balance Criteria. In Frank Dehne, Jorg-Rudiger Sack, and Nicola Santoro, editors, 1st International Workshop on Algorithms and Data Structures, Lecture Notes in Computer Science, volume 382, pages 393--402. Springer-Verlag, 1989.

[A, 1999] A. Andersson. General Balanced Trees, Journal of Algorithms 30, 1-18, (1999).

[R, 1966] H. J. Ryser. Combinatorial Mathematics, 1966.

[S, 1986] H. A. Sahakyan. Greedy Algorithms for Synthesis of (0,1) Matrices, Doklady Akademii Nauk Arm. SSR, v. 83, 5, pp. 207-209 (1986).

[S, 1995] H. Sahakyan. Hierarchical Procedures with the Additional Constraints, II Russia, with participation of NIS Conference, Pattern Recognition and Image Analysis: New Information Technologies, Ulianovsk, 1995, pp.76-78.

Author information

Hasmik Sahakyan - Institute for Informatics and Automation Problems, NAS Armenia, P.Sevak St. 1, Yerevan-14, Armenia; e-mail: hasmik@ipia.sci.am

Levon Aslanyan – Institute for Informatics and Automation Problems, NAS Armenia, P.Sevak St. 1, Yerevan-14, Armenia; e-mail: lasl@sci.am