Hierarchical Clustering: a 0.585 Revenue Approximation

Noga Alon* Yossi Azar† Danny Vainstein‡

June 2, 2020

Abstract

Hierarchical Clustering trees have been widely accepted as a useful form of clustering data, resulting in a prevalence of adopting fields including phylogenetics, image analysis, bioinformatics and more. Recently, Dasgupta (STOC 16') initiated the analysis of these types of algorithms through the lenses of approximation. Later, the dual problem was considered by Moseley and Wang (NIPS 17') dubbing it the Revenue goal function. In this problem, given a nonnegative weight w_{ij} for each pair $i, j \in [n] = \{1, 2, \ldots, n\}$, the objective is to find a tree T whose set of leaves is $[n]$ that maximizes the function $\sum_{i<j\in[n]} w_{ij}(n - |T_{ij}|)$, where $|T_{ij}|$ is the number of leaves in the subtree rooted at the least common ancestor of i and j.

In our work we consider the revenue goal function and prove the following results. First, we prove the existence of a bisection (i.e., a tree of depth 2 in which the root has two children, each being a parent of $n/2$ leaves) which approximates the general optimal tree solution up to a factor of $\frac{1}{2}$ (which is tight). Second, we apply this result in order to prove a $\frac{2}{3}p$ approximation for the general revenue problem, where p is defined as the approximation ratio of the Max-Uncut Bisection problem. Since p is known to be at least 0.8776 [3] [18], we get a 0.585 approximation algorithm for the revenue problem. This improves a sequence of earlier results which culminated in an 0.4246-approximation guarantee [1].

1 Introduction

The notion of Hierarchical Clustering (HC) trees has been introduced and subsequently studied for several decades. The notion was first considered due to its applications to the realm of phylogenetics [16] [14]. Here, given genomic similarities between species our goal is to hierarchically cluster the species in a way that captures fine-grained relations between different species. Since then, the notion of HC trees has expanded to many additional fields. See [6] for a survey on the subject.

Typically, schemes for generating HC trees fall into one of two categories: Agglomerative algorithms (i.e., bottom-up) and Divisive algorithms (i.e., top-down). Agglomerative algorithms initially start with a partition in which each data point forms its own set. The algorithm then proceeds to recursively merge different sets, terminating once all data points are contained in the same set. Notably, the well-studied average linkage algorithm is an example of such a procedure in which in each step two sets maximizing the average induced weight are merged. On the other hand, divisive algorithms start with a single set containing all the data points, and then proceed to recursively split sets, terminating once each data point remains alone in its set.

[*] Princeton University and Tel Aviv University. Email: nalon@math.princeton.edu. Research supported in part by NSF grant DMS-1855464, BSF grant 2018267 and the Simons Foundation.

[†] School of Computer Science, Tel-Aviv University. Email: azar@tau.ac.il

[‡] School of Computer Science, Tel-Aviv University. Email: dannyvainstein@gmail.com

Recently, Dasgupta [13] formally defined the notion of a "good" HC tree. With this definition he elegantly bridged the gap between HC trees and the field of approximation algorithms by defining a minimization cost function (see related work for an extended discussion on this goal function). Thereafter, Moseley and Wang [15] considered the complementary maximization variant of this problem - namely, the revenue goal function.

Both problems considered the following model. Assume we are given a set of n data points, V with some notion of similarity between them. The similarity is formally captured through similarity-edges between any two data points, represented by a weighted graph $G = (V, E, w)$ with $w : E \to \mathbb{R}^+$. Alternatively, G may be viewed as a complete graph in which $w(e) = 0$ for any $e \notin E$. Our goal is then to construct an HC tree, T, such that its leaves are in a 1-1 correspondence with our data points, V. Note that given such a tree, every internal node represents a set of data points, which are the leaves of its subtree, and a partition of this set given by the sets of leaves of the subtrees rooted at the children of the node.

Intuitively speaking, since higher weighted edges correspond to more similar data points, it is desirable to split the endpoints of such edges low in a good HC tree. Formally, Moseley and Wang [15] defined the revenue problem as,

$$\max_T R_G(T) = \max_T \textstyle\sum_{e=\{ij\}\in E} w_{ij}(n - |T_{ij}|),$$

where T is an HC tree, T_{ij} denotes the subtree rooted at the least-common-ancestor (LCA) of data points i and j and $|T_{ij}|$ denotes the amount of data points in T_{ij}.

In [15], Moseley and Wang considered several algorithms. Notably, they considered the random algorithm, that simply splits data points randomly at every cut (henceforth denoted by $RAND$), and the average-linkage algorithm. They showed that both yield an approximation factor of 1/3. Subsequently, Charikar et al. [8] showed that one can beat average-linkage through the use of semi-definite programming, improving the bound to 0.3364. Recently, Ahmadian et al. [1] managed to leverage the Max-Uncut Bisection (MUB) problem in order to prove a 0.4246 approximation. In our paper we improve upon this result and show an improved approximation of 0.585.

Our contributions. We consider the revenue goal function and prove the following results.

- We show that for any revenue instance, there exists a bisection, X, that is, a tree of depth 2 in which the root has two children, each being the parent of $n/2$ leaves, such that $R(X) \geq \frac{1}{2}OPT$, where OPT denotes the revenue gained by the optimal tree (see Theorem 3.1). In order to show such existence we make use of two random processes: we randomly fix the order of the leaves in the optimal tree in an appropriate way, and then randomly generate our bisection, X. We emphasize the fact that even though OPT makes use of an arbitrarily deep tree, it is enough to consider a single cut in order to gain half the revenue.

- Using our result regarding the existence of a large revenue generating bisection, we prove a 0.585 approximation for the revenue problem. We note that in fact we show a $\frac{2}{3}p$ approximation where $p = 0.8776$ is the best known approximation for the MUB problem.

Remark. *The algorithm we consider is that which solves the MUB problem for the first cut and then proceeds using the random algorithm. In [1], Ahmadian et al. considered this algorithm coupled with the random algorithm - they used both algorithms simultaneously while taking the maximal revenue of the two. They showed that this results in a 0.4246 approximation with respect to the gained revenue. Somewhat surprisingly we show that the former algorithm (MUB and then random) on its own is enough to yield an approximation of 0.585.*

Techniques. Our first result makes use of a new upper bound on the optimal solution, which may be of independent interest. Specifically, given an optimal solution, we embed its leaves on a line such that its root is above the line and we have no resulting crossing edges. This clearly yields an ordering of the leaves (see Figure 1). We then consider the distance between any two data points, i and j within this ordering (simply the difference in rank) and observe that this is in fact a lower bound on $|T_{ij}|$. This in turn yields an upper bound on the optimal solution.

Next, we make use of this bound by randomly generating a bisection that gains revenue that is "large" with respect to the bound and by showing that in expectation this bound is "far" from the optimal solution. Both "large" and "far" will be formally defined later on.

Related work. Dasgupta [13] kicked off the line of work considering HC trees within the realm of approximation algorithms. In his paper, he considered similarity-edges and defined the cost of an HC tree as,

$$\min_T C_G(T) = \min_T \sum_{e=\{ij\}\in E} w_{ij}|T_{ij}|.$$

Note that the revenue goal function is in fact complementary to that of Dasgupta's function (that is, the optimal solution is the same for both, albeit with different goal function values).

In [13], many general properties pertaining to this goal function were discussed. Notably, it was shown that this goal function is intuitive in that (1) on complete graphs (with no structure) all HC trees yield the same cost, (2) on disconnected graphs, optimal HC trees begin by splitting disconnected components and (3) the goal function is modular. He further presented an $O(\log^{1.5}(n))$ approximation algorithm via recursive sparsest cut. Later, both Charikar and Chatziafratis [7] and Cohen-Addad et al. [12] showed that this algorithm is in fact an $O(\sqrt{\log n})$ approximation. In the hardness domain, Dasgupta [13] showed that the problem is NP-hard via a reduction to a variant of the NAE-SAT problem. This was later improved by Charikar and Chatziafratis [7], showing that in fact no constant approximation exists (assuming the Small Set Expansion hypothesis). Cohen-Addad et al. [11] managed to overcome the latter worst-case specific result by considering average case inputs defined by a stochastic block model and its hierarchical extension. Here, they managed to establish an $O(1)$ approximation.

Following Dasgupta's work, Cohen-Addad et al. [12] considered the case of dissimilarity-edges. In this case, Dasgupta's cost function is now translated to a maximization problem. Given an HC tree, T, we now denote its gained value as its gained dissimilarity, $D_G(T)$. In this setting, both the random algorithm ($RAND$) and the average-linkage algorithm yield dissimilarity values of $2/3$ of the optimal solution. Charikar et al. [8] improved upon this by proving an approximation of 0.6671 which makes use of a more delicate multi-phase algorithm.

Several other extensions to the formerly defined HC goal functions were also considered. One such extension is that of structural constraints. Specifically, every constraint appears in the form of "$i, j|k$" for data points i, j and k. A constraint is then considered satisfied if $k \notin T_{ij}$, for an HC tree, T. Aho et al. [2] considered this problem in the phylogenetic realm where this notion gives rise to the problem of constructing a phylogenetic tree that satisfies a set of lineage constraints on common ancestors. This notion has more recently been investigated in the domain of HC, by Chatziafratis et al. [10]. In their paper they extended Dasgupta's goal function to include structural constraints and showed an $O(k\sqrt{\log n})$ approximation where k is the number of constraints. For additional works in the realm of HC trees see [5], [4], [17], [9].

2 Notation and Preliminaries

We introduce several definitions that will aid us throughout the following sections.

Definition 1. *Given a revenue instance $G = (V, E, w)$, we denote the optimal revenue tree by OPT.*

We shall abuse notation and refer both to the optimal revenue tree and to its generated revenue as OPT (it will be clear from context which of these definitions we will be referring to).

Definition 2. *Given an HC tree T, we denote by $R(T)$ the revenue it yields and for any similarity edge e, we denote by $R_T(e)$ the revenue gained by the edge e with respect to the HC tree, T.*

The following definition will be useful when considering the revenue generated with respect to some similarity edge.

Definition 3. *Given a revenue instance $G = (V, E, w)$ and a similarity edge $e = \{i, j\}$ we denote $T_e = T_{ij}$.*

Note that therefore, for any HC tree T, $R_T(e) = w_e(n - |T_e|)$, where $|T_e|$ is the number of leaves in the tree T_e. Further note that with these definitions we may assume w.l.o.g. that any optimal tree is binary.

Finally, to ease notation we shall refer to the data points of a given revenue instance $G = (V, E, w)$, as $V = \{1, 2, \ldots, n\}$.

3 Existence of a High Revenue Bisection

Theorem 3.1. *For any revenue instance and corresponding optimal solution, OPT, there exists a bisection, X, satisfying*

$$R(X) \geq \frac{1}{2} OPT,$$

and this is asymptotically tight.

Before we prove the theorem, we introduce some notation. To simplify the presentation we assume from now on that the number n of leaves is even.

Defining a tree ordering: Given an HC tree T, we may fix its leaves in several different orders. Each order is produced by representing T as a planar graph where its leaves are all embedded on a line, the root is above the line and all edges of the tree are straight lines going down from each parent to its children with no crossing edges. Denote each such ordering by the function $\pi(T) : V \to [n]$.

Henceforth we will consider π with respect to the optimal solution OPT and denote $\pi = \pi(OPT)$. Recall that we may assume w.l.o.g. that OPT is a binary tree. Next we define a distribution over all feasible orderings, π.

Definition 4. *Let P_π denote the distribution generated by choosing uniformly at random a subset of internal nodes from OPT and swapping the placements of the left and right subtrees of each of these chosen nodes.*

Finally, given an ordering π, and an edge $e = \{i, j\}$, let $y_e^\pi = |\pi(i) - \pi(j)|$, denote the distance between leaves i and j in the fixed tree, OPT. For a pictorial example, see Figure 1.

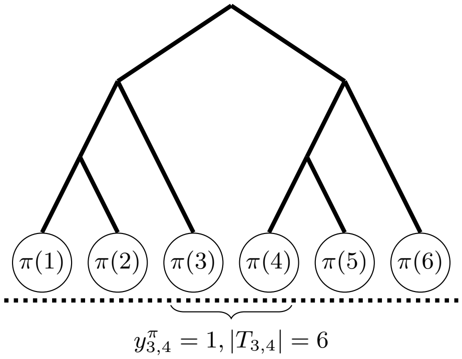

Figure 1: Embedding an HC tree with order π.

In order to prove Theorem 3.1 we first upper bound the revenue gained by OPT.

Observation 1. *$\forall \pi, \forall e \in E : y^{\pi}_e \leq |T_e|$, where both π and T_e are defined with respect to OPT.*

In itself, this will not result in a good enough bound on OPT. Therefore, we show that on average, y^{π}_e is far from T_e.

Lemma 3.2. *$E_{P_\pi}[y^{\pi}_e] = \frac{|T_e|}{2}$.*

Proof. Let $e = \{i, j\}$. It can be shown through simple induction that for any tree with n leaves, and for any leaf, k, $E_{P_\pi}[\pi(k)] = \frac{n+1}{2}$. (This also follows by linearity of expectation from the simple fact that for every two distinct leaves i and j the probability that $\pi(i) < \pi(j)$ is exactly 1/2.)

Denote by n_i and n_j the number of leaves in the subtrees containing i and j, each rooted at a separate child of i and j's least-common-ancestor. By the definition of P_π, the probability that i (together with the n_i leaves of its subtree) appears before j (and the n_j leaves in its subtree) is exactly 1/2, resulting in:

$$E_{P_\pi}[y^{\pi}_e] = (1/2)(n_i + \frac{n_j+1}{2} - \frac{n_i+1}{2}) + (1/2)(n_j + \frac{n_i+1}{2} - \frac{n_j+1}{2}) = \frac{|T_e|}{2}.$$

□

Let $Y_\pi = \sum_{e \in E} w_e \cdot y^{\pi}_e$. By linearity of expectation we get the following lemma.

Lemma 3.3. *There exists an ordering of OPT, π^*, such that,*

$$Y_{\pi^*} \leq \sum_{e \in E} w_e \cdot \frac{|T_e|}{2}.$$

Proof. By linearity of expectation and Lemma 3.2,

$$E_{P_\pi}[Y_\pi] = \sum_{e \in E} w_e \cdot E_{P_\pi}[y^{\pi}_e] = \sum_{e \in E} w_e \cdot \frac{|T_e|}{2}.$$

Therefore, there exists an ordering as needed. □

Next we show that there exists a distribution over the set of all bisections with high (to some degree) revenue with respect to the revenue gained by considering y^{π}_e rather than $|T_e|$.

Lemma 3.4. *Given any ordering of OPT, π, there exists a distribution, P_X, over all bisections, X, such that for any edge $e \in E$,*

$$E_{P_X}[R_X(e)] \geq \frac{1}{2} w_e(n - 2y_e^\pi).$$

Proof. Fix π. We relabel the leaves of OPT such that $\pi(i) = i$ (this is simply to ease the notation). Next we define a distribution over all bisections, X. Consider the following random process: choose x uniformly at random from $\{1, \ldots, \frac{n}{2}\}$. Then define X to be the bisection, $\{x, x+1, \ldots, x + \frac{n}{2} - 1\}, \{1, 2, \ldots, x-1, x + \frac{n}{2}, x + \frac{n}{2} + 1, \ldots, n\}$.

Now consider some edge, $e \in E$. If $y_e^\pi \geq \frac{n}{2}$, then $n - 2y_e^\pi \leq 0$. $R_X(e)$ is always nonnegative (since it is defined as the revenue gained by e) and thus the assertion of the lemma holds in this case.

Otherwise, $y_e^\pi \leq \frac{n}{2} - 1$ for $e = \{i, j\}$. In this case, the probability that i and j are cut by the bisection is exactly $\frac{y_e^\pi}{n/2} = \frac{2y_e^\pi}{n}$. Furthermore, since X is a bisection, any uncut edge yields a revenue of $n/2$. Overall,

$$E_{P_X}[R_X(e)] \geq w_e(1 - \frac{2y_e^\pi}{n})\frac{n}{2},$$

completing the proof of the lemma. □

Recall that by the definition of $R(X)$ it follows that $R(X) = \sum_{e \in E} R_X(e)$. Therefore, by combining lemmas 3.3 and 3.4, we may sum over all edges and get the following observation.

Observation 2. *Given the ordering, π^*, guaranteed by Lemma 3.3, we get,*

$$E_{P_X}[R(X)] \geq \sum_{e \in E} \frac{1}{2} w_e(n - 2y_e^{\pi^*}).$$

We are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. Fix an ordering of OPT to be π^* as guaranteed by Lemma 3.3. By Observation 2, there exists a bisection X^* satisfying

$$R(X^*) \geq E_{P_X}[R(X)] \geq \sum_{e \in E} \frac{1}{2} w_e(n - 2y_e^{\pi^*}).$$

Thus, by Lemma 3.3,

$$\begin{aligned}
R(X^*) &\geq \sum_{e \in E} \frac{1}{2} w_e(n - 2y_e^{\pi^*}) \\
&= \sum_{e \in E} \frac{1}{2} w_e(n) - Y_{\pi^*} \\
&\geq \sum_{e \in E} \frac{1}{2} w_e(n) - \sum_{e \in E} w_e \cdot \frac{|T_e|}{2} \\
&= \frac{1}{2} \sum_{e \in E} w_e(n - |T_e|) \\
&= \frac{1}{2} OPT.
\end{aligned}$$

To show that the result is asymptotically tight, consider the instance $G = (V, E)$ on $|V| = n$ vertices which is a matching of $n/2$ edges, each having weight 1. In this case, one solution is a binary tree in which if $\{i, j\} \in E$ then the LCA of i and j (as defined by the solution) is i and j's immediate parent. This solution generates $n-2$ revenue for every edge of the matching guaranteeing that $OPT \geq \frac{n}{2}(n-2)$.

On the hand, consider an arbitrary tree of depth 2 such that its first cut is a bisection and denote it by T. Let w_1 and w_2 denote the total weight of edges in each set generated by the cut. Therefore, $w_1 + w_2 \leq \frac{n}{2}$. The revenue generated from T is exactly $\frac{n}{2}w_1 + \frac{n}{2}w_2 \leq \frac{n^2}{4}$. Overall,

$$R(T) \leq (\frac{1}{2} + o(1))OPT.$$

□

Remark. *Note that Theorem 2 may be derandomized in the sense that given any binary HC tree we may produce a bisection with at least half its revenue deterministically and efficiently. Indeed, this can be done using the method of conditional expectations. We omit the (simple) details, since although this implies that given an optimal HC tree one can obtain a bisection with revenue at least OPT/2 deterministically and efficiently, this does not yield any new algorithmic consequence when an optimal HC tree is not given.*

4 A 0.585 Approximation for the Revenue Goal Function

We define the approximation algorithm as a 2 step process: first we cut all data points using some black box algorithm that produces a bisection, denoted henceforth as ALG_{MUB}. Thereafter, we continue splitting each cluster randomly. Denote the combined algorithm by ALG.

In order to show an approximation bound for ALG, we first need to consider the MUB problem. In this problem we are given a weighted graph and our goal is to create a bisection maximizing the weights of uncut edges. We note that if we restrict ourselves to revenue trees which are bisections (i.e., the first cut splits the data points into two equal sets, then each set is cut only once using a "star" subtree - see Figure 2), the MUB optimal solution and the revenue optimal solution are in fact the same.

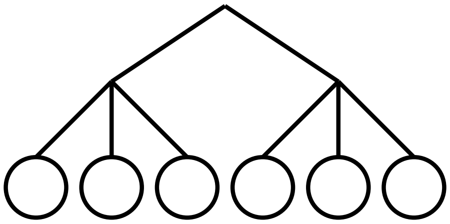

Figure 2: Example of a bisection revenue tree - revenue is only gained through edges uncut by the first cut.

Let p denote the approximation ratio algorithm ALG_{MUB} guarantees for the MUB problem. Note that p is at least 0.8776 (see [3], [18]). Thus, by defining ALG_{MUB} to be such an algorithm, the following theorem shows that algorithm ALG is a 0.585 approximation for the revenue problem.

Theorem 4.1. *Algorithm ALG guarantees an approximation ratio of $\frac{2}{3}p = 0.585$ for the revenue problem, where p is defined as the approximation ratio for the MUB problem.*

Proof. Let T_{ALG} denote the tree generated by algorithm ALG. It is a known simple fact that the random algorithm generates a revenue of at least $\frac{1}{3}nw_e$ for any edge e. Therefore, if we denote by W_L and W_R the weight of the uncut edges generated by the first cut of our algorithm, then,

$$R(T_{\text{ALG}}) \geq W_L(\frac{n}{2} + \frac{1}{3} \cdot \frac{n}{2}) + W_R(\frac{n}{2} + \frac{1}{3} \cdot \frac{n}{2}) = (W_L + W_R)\frac{2n}{3}.$$

Denote by W_{L^*} and W_{R^*} the weights of the uncut edges generated by the optimal MUB solution and let p denote our first cut's approximation with respect to the MUB problem. Furthermore, let X^* denote the optimal solution to the revenue problem restricted to bisections. As noted earlier, X^* corresponds to W_{L^*} and W_{R^*}. Therefore,

$$W_L + W_R \geq p(W_{L^*} + W_{R^*}) = \frac{2p}{n}R(X^*).$$

Let OPT denote the revenue gained by the optimal solution. Thus, leveraging Theorem 3.1 yields,

$$\begin{aligned} R(T_{\text{ALG}}) \geq & \frac{2n}{3}(W_L + W_R) \geq \\ & \frac{4p}{3}(R(X^*)) \geq \\ & \frac{2p}{3}OPT. \end{aligned}$$

Since p is known to be at least 0.8776, we get that ALG is a $\frac{2}{3}p = 0.585$ approximation algorithm. □

References

[1] Sara Ahmadian, Vaggos Chatziafratis, Alessandro Epasto, Euiwoong Lee, Mohammad Mahdian, Konstantin Makarychev, and Grigory Yaroslavtsev. Bisect and conquer: Hierarchical clustering via max-uncut bisection. *CoRR*, abs/1912.06983, 2019.

[2] Alfred V. Aho, Yehoshua Sagiv, Thomas G. Szymanski, and Jeffrey D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM Journal on Computing*, 10(3):405–421, 1981.

[3] Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. Better balance by being biased: A 0.8776-approximation for max bisection. *ACM Trans. Algorithms*, 13(1):2:1–2:27, 2016.

[4] Pranjal Awasthi, Maria Florina Balcan, and Konstantin Voevodski. Local algorithms for interactive clustering. *The Journal of Machine Learning Research*, 18(1):75–109, 2017.

[5] Maria-Florina Balcan and Avrim Blum. Clustering with interactive feedback. In *International Conference on Algorithmic Learning Theory*, pages 316–328. Springer, 2008.

[6] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping Multidimensional Data - Recent Advances in Clustering*, pages 25–71. 2006.

[7] Moses Charikar and Vaggos Chatziafratis. Approximate hierarchical clustering via sparsest cut and spreading metrics. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 841–854, 2017.

[8] Moses Charikar, Vaggos Chatziafratis, and Rad Niazadeh. Hierarchical clustering better than average-linkage. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2291–2304, 2019.

[9] Moses Charikar, Vaggos Chatziafratis, Rad Niazadeh, and Grigory Yaroslavtsev. Hierarchical clustering for euclidean data. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 2721–2730, 2019.

[10] Vaggos Chatziafratis, Rad Niazadeh, and Moses Charikar. Hierarchical clustering with structural constraints. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 773–782, 2018.

[11] Vincent Cohen-Addad, Varun Kanade, and Frederik Mallmann-Trenn. Hierarchical clustering beyond the worst-case. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6201–6209, 2017.

[12] Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. Hierarchical clustering: Objective functions and algorithms. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 378–397, 2018.

[13] Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 118–127, 2016.

[14] N Jardine and R Sibson. A model for taxonomy. *Mathematical Biosciences*, 2(3-4):465–482, 1968.

[15] Benjamin Moseley and Joshua Wang. Approximation bounds for hierarchical clustering: Average linkage, bisecting k-means, and local search. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 3094–3103, 2017.

[16] Peter HA Sneath and Robert R Sokal. Numerical taxonomy. *Nature*, 193(4818):855–860, 1962.

[17] Sharad Vikram and Sanjoy Dasgupta. Interactive bayesian hierarchical clustering. In *International Conference on Machine Learning*, pages 2081–2090, 2016.

[18] Chenchen Wu, Donglei Du, and Dachuan Xu. An improved semidefinite programming hierarchies rounding approximation algorithm for maximum graph bisection problems. *J. Comb. Optim.*, 29(1):53–66, 2015.