The Directed Circular Arrangement Problem*

Joseph (Seffi) Naor† Roy Schwartz‡

Abstract

We consider the problem of embedding a directed graph onto evenly-spaced points on a circle while minimizing the total weighted edge length. We present the first poly-logarithmic approximation factor algorithm for this problem which yields an approximation factor of $O(\log n \log\log n)$, thus improving the previous $\tilde{O}(\sqrt{n})$ approximation factor. In order to achieve this, we introduce a new problem which we call the *directed penalized linear arrangement*. This problem generalizes both the directed feedback edge set problem and the directed linear arrangement problem. We present an $O(\log n \log\log n)$ approximation factor algorithm for this newly defined problem. Our solution uses two distinct directed metrics ("right" and "left") which together yield a lower bound on the value of an optimal solution. In addition, we define a sequence of new directed spreading metrics that are used for applying the algorithm recursively on smaller subgraphs. The new spreading metrics allow us to define an asymmetric region growing procedure that accounts simultaneously for both incoming and outgoing edges. To the best of our knowledge, this is the first time that a region growing procedure is defined in directed graphs that allows for such an accounting.

1 Introduction

We consider in this paper the *directed circular arrangement* (DCA) problem. The input is a directed graph with non-negative weights on the edges. The goal is to embed the graph onto evenly-spaced points on a circle, such that all edges are oriented in the same direction, say, *clockwise*, and the weighted sum of the lengths of the edges is minimized. The length of an edge is defined to be the (clockwise) distance between its two endpoints in the embedding. We note that each vertex is embedded onto a distinct point. The directed circular arrangement problem was recently considered by Liberatore [10] who showed that the problem is NP-hard and gave an $\tilde{O}(\sqrt{n})$-approximation factor algorithm. Liberatore [10] also considered embedding an undirected graph onto a circle while minimizing the total weighted edge length. The length of each edge (u, v) is then defined to be the shortest distance between u and v on the circle. For this version of the problem, an $O(\log n)$-approximation algorithm was given in [10]. We note that this factor is improved (implicitly) to $O(\sqrt{\log n}\log\log n)$ by later works [3, 7], which improve the approximation factor of undirected linear arrangement.

*A preliminary version of the paper appeared in the Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, Louisiana (2004), pp 86-95.

†Computer Science Department, Technion, Haifa 32000, Israel. E-mail: *naor@cs.technion.ac.il*. Research supported in part by a US-Israel BSF grant, and by EU contract IST-1999-14084 (APPOL II).

‡Computer Science Department, Technion, Haifa 32000, Israel. E-mail: *schwartz@cs.technion.ac.il*.

One can see the resemblance between circular arrangements and linear arrangements. Linear arrangement is a classical NP-hard problem which has been studied extensively starting with the work of Leighton and Rao [9]. The goal is to embed an undirected graph onto evenly-spaced points on a line while minimizing the total weighted edge length. In the directed version of this problem, it is required that all the edges go in only one direction, implying that the graph has to be acyclic. (Unlike the case of a directed circular arrangement.) Rao and Richa [11] gave an $O(\log n)$-approximation algorithm for both the undirected and directed linear arrangement problems using the spreading metric technique developed by Even et al. [4]. Charikar et al. [3] and independently Feige and Lee [7] improved the approximation factor to $O(\sqrt{\log n}\log\log n)$ by extending the spreading metric technique to ℓ_2^2.

Developing approximation algorithms for the directed circular arrangement problem presents two main obstacles. First, applying divide and conquer techniques usually requires breaking an instance into smaller instances of the same type. There is no obvious way to do that in the case of circular arrangements. Moreover, there is an inherent asymmetry in the problem which is manifested in the fact that if we have a subgraph of k vertices which are placed next to each other in a solution, then some of the edges that connect vertices in this component may have a large length which does not depend on k. (See Figure 1, where the edge $(u \to v)$ has length of at most $O(k)$ and the edge $(v \to u)$ has length of at least $\Omega(n-k)$. If $k = o(n)$ then the latter is at least $\Omega(n)$ which is independent of k). Second, it does not seem that the spreading-metric based techniques of [4, 11] can be applied here in a straightforward way. We note that a spreading metric can be defined for the directed circular arrangement problem. However, since the graph is directed, rounding it via region growing as in [8, 12] only allows bounding the cost of cut edges in one direction (either incoming edges or outgoing edges), while it is necessary to bound the cost of edges going in both directions.

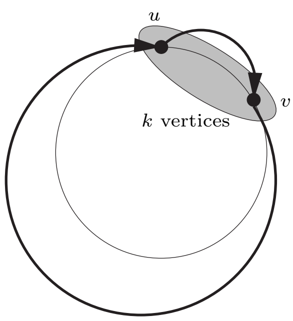

Figure 1: Asymmetry of directed circular arrangement, $(u \to v)$ has length of at most $O(k)$ and $(v \to u)$ has length of at least $\Omega(n-k)$.

We now elaborate on several applications of the directed circular arrangement problem. Scalability is a fundamental issue in server design and it requires the ability to support a large number of requests for data items. This can be implemented by multicasting the data items periodically (see Liberatore [10]) leading to a scheduling problem where the goal is to find the optimal order in a period. Each period is a permutation of the data items which are modeled as vertices in a directed graph with weighted edges connecting them. Each directed edge represents a dependency between two data items and the weight of the edge corresponds to the strength of the dependency. (See also [2].) Thus, we can think of the permutation as an embedding of a directed graph onto the circle while minimizing the weighted sum of the lengths of the edges.

A different problem arises in the context of load minimization in (clockwise) directed ring networks. Assume that there are n stations connected to a central hub in a star topology. The hub is capable of connecting any two stations and the connections through the hub are directed. Thus, if station A is placed in position i in the ring, and station B is placed in position j, then a message that is sent from A to B has to go through the hub $(j-i) \mod n$ times (the clockwise distance from A to B). An $n \times n$ load matrix is given, specifying the expected load between any pair of stations. Thus, the total load on the hub, summed over all pairs of stations A and B, is the expected load between A and B multiplied by the clockwise distance between them. Therefore, we get an instance of the directed circular arrangement problem.

2 Our Results

Our main result is an $O(\log n \log \log n)$-approximation algorithm for the directed circular arrangement problem. We obtain this result by defining a new problem, the *directed penalized linear arrangement* problem, and use a solution to this newly defined problem to approximate the directed circular arrangement.

The directed penalized linear arrangement problem takes as input a directed graph and embeds it onto evenly spaced points on a straight line. In the embedding, edges that are directed to the right contribute to the objective function a term which is proportional to their (weighted) length, while edges that are directed to the left contribute to the objective function a term which is only proportional to the weight of the edge. We think of the latter term as a *penalty*.

The main contribution of this paper is an $O(\log n \log \log n)$-approximation algorithm for the directed penalized linear arrangement problem. Our solution uses two distinct directed metrics ("right" and "left") which together yield a lower bound on the value of an optimal solution. In addition, we define a sequence of new directed spreading metrics that are used for applying the algorithm recursively on smaller subgraphs. The directed penalized linear arrangement problem still shares some of the asymmetry characteristics of the directed circular arrangement problem. However, the new spreading metrics allow us to define an asymmetric region growing procedure that accounts simultaneously for both incoming and outgoing edges. To the best of our knowledge, this is the first time that a region growing procedure in directed graphs is defined in a way that allows for such an accounting.

We note that the directed penalized linear arrangement problem generalizes both the directed feedback edge set problem and the directed linear arrangement problem (also known as minimum storage-time product for the special case where the time required for each task is exactly 1). The best known approximation for directed feedback edge set is $O(\log n \log \log n)$ [6, 12]. Thus, our approximation algorithm for the directed penalized linear arrangement matches the best known approximation factor for the directed feedback edge set.

Finally, we present a very simple extension of existing approximation algorithm designed for ordering problems which approximates the undirected circular arrangement problem, therefore simplifying the algorithm of Liberatore [10]. Our algorithm achieves an approximation factor of $O(\sqrt{\log n}\log\log n)$ by the results of [3, 7]. Additionally, we show a connection between undirected linear and circular arrangements. We prove that if there is an α-approximation for undirected circular arrangement, then there is a 2α-approximation for undirected linear arrangement.

3 Preliminaries

Our input is a directed graph $G = (V, E)$ and a non-negative edge weight function $w : E \to R^+$. Let $|V| = n$, and let q and p denote non-negative *penalties* which will be determined later.

The *directed circular-arrangement* (DCA) problem is defined as follows. The goal is to find an embedding f of the vertex set onto n evenly-spaced points on a circle, such that all edges are oriented in the same direction, say, clockwise, and the weighted sum of the lengths of the edges is minimized. That is, we need to find a one-to-one function $f : V \to \{1, 2, \ldots, n\}$ which minimizes:

$$\sum_{(u,v)\in E} w(u,v)\Big(\big(f(v) - f(u)\big) \mod n\Big).$$

The DCA problem is NP-hard, as proved in [10].

The *directed linear arrangement* (DLA) problem is defined as follows. Assume the graph G is *acyclic*. The goal is to find an embedding f of the vertex set V onto n evenly-spaced points on a line, such that if $(u, v) \in E$, then $f(u) < f(v)$, and the weighted sum of the lengths of the edges is minimized. That is, we need to find a topological ordering $f : V \to \{1, 2, \ldots, n\}$ which minimizes:

$$\sum_{(u,v)\in E} w(u,v)(f(v) - f(u)).$$

The *directed feedback edge set* (DFES) problem is defined as follows. The goal is to find a minimum weight set of edges that intersects every directed cycle in G. That is, we need to find a subset of edges $X \subseteq E$ such that for every directed cycle C in G, $C \cap X \neq \emptyset$, minimizing

$$\sum_{(u,v)\in X} w(u,v).$$

The *(p,q)-directed penalized linear arrangement* (DPLA) problem is defined as follows. The goal is to find an embedding f of the vertex set onto n evenly-spaced points on a line. Each edge can be oriented in one of two directions, which for convenience we call *right* and *left*. If an edge is oriented to the right, then its contribution to the objective function is q times its weighted length in the embedding. If an edge goes to the left, then its contribution to the objective function is p times its weight. Thus, our goal is to find a one-to-one function $f : V \to \{1, 2, \ldots n\}$ which minimizes

$$\sum_{(u,v)\in E} w(u,v)h(u,v),$$

where $h(u, v) = q(f(v) - f(u))$ if $(u \to v)$ goes to the right (i.e. $f(v) > f(u)$), or $h(u, v) = p$ if $(u \to v)$ goes to the left (i.e. $f(v) < f(u)$). We call edges that go to the right *non-penalized* edges and edges that go to the left *penalized* edges.

The DLA problem is in fact a special case of the DPLA problem. Consider an acyclic graph G with weight function w and set $q = 1$ and $p = \infty$. Without loss of generality, $w(e)$ is finite for all $e \in E$. Clearly, an optimal solution for the DPLA problem in this case is also an optimal solution for the DLA problem, since any edge oriented to the left would yield a solution of infinite value to the DPLA problem. (However, a solution of finite value to the DPLA problem obviously exists.) The NP-hardness of the DLA problem implies that DPLA is also NP-hard.

The DFES problem is also a special case of the DPLA problem. Consider a directed graph G with weight function w and set $q = 0$ and $p = 1$. Clearly, an optimal solution for the DPLA problem in this case is also an optimal solution for the DFES problem, since all edges oriented to the left in the arrangement are a legal feedback edge set of G. The best known approximation factor for DFES is $O(\log n \log \log n)$, and it follows from the work of Seymour [12] . We present an $O(\log n \log \log n)$ approximation for DPLA, thus matching the approximation factor known for DFES.

In this paper we use *directed metrics*. We will use the following definition throughout the rest of the paper.

Definition 1. (X, d) *is a* directed metric *if the function* $d : X \times X \to R^+$ *satisfies the following conditions:*

- $d(x, x) = 0$, $\forall x \in X$
- $d(x, y) + d(y, z) \geq d(x, z)$, $\forall x, y, z \in X$

Note that a directed distance function d might not be symmetric. Therefore, one can only use the directed version of the triangle inequality.

We need the definition of a *separator* in an undirected graph. Let G be an undirected graph with a non-negative edge weight function $w : E \to R^+$ and let ρ be a parameter such that $0 < \rho \leq 1$. A ρ-separator is a subset of edges whose removal partitions the graph into connected components such that the number of vertices in each component is at most $\rho|V|$. The goal is to find a minimum weight ρ-separator. This problem was considered by Leighton and Rao [9] who provided the first logarithmic (pseudo) approximation factor for constant values of ρ. The best approximation factor known is due to Even, Naor, Rao and Schieber [5]. They find a ρ'-separator, where $\rho' > \rho$, of weight at most $\left(\frac{\rho'}{\rho'-\rho} + o(1)\right) \ln n$ times an optimal ρ-separator. We call such a separator a (ρ, ρ')-separator.

We denote by $OPT_{\Pi}(G)$ the value of an optimal solution to problem Π on input graph G.

4 An Approximation Algorithm for Directed Circular Arrangement

The idea of the algorithm is to "break" the graph into small enough components and arrange the vertices of each component next to each other. We say that $(u \to v)$ crosses other components if $\big(f(u) - f(v)\big) \mod n < \big(f(v) - f(u)\big) \mod n$. When arranging each component, we know that edges that need to cross other components are going to be long. This suggests using on each component the directed penalized linear arrangement approximation algorithm, where the *right* punishment, q, is 1 and the *left* punishment, p, is determined by the lower bound on the length of such edges. We can state the approximation algorithm for DCA:

APPROXDCA($G = (V, E)$)

1. Remove edge direction and find a $(\frac{1}{3}, \frac{1}{2})$-separator in G.
   Let $G_i = (V_i, E_i)$, $i = 1, 2, \ldots, k$, be the resulting components.

2. Find for each component G_i an approximate DPLA solution with
   $q_i = 1$ and $p_i = |V \setminus V_i|$.

3. Arrange the vertices of each component G_i next to each other on the cycle
   according to the order of the approximate DPLA solution.
   (the order between the components is arbitrary).

4. Output this arrangement.

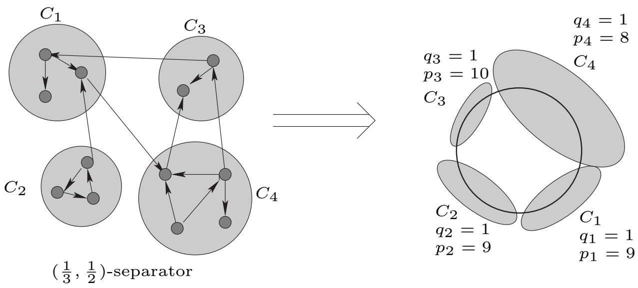

Figure 2: Algorithm **APPROXDCA**

Note that in Step (1), since a separator is defined for undirected graphs, one needs to remove edge direction in order to find a separator. Thus, the weight of an undirected edge (u, v) after removing edge direction, is the sum of weights of *all* directed edges from u to v and *all* directed edges from v to u. We note that the first part of the proof of the following theorem (bound on the arrangement cost of the separator edges) is similar to a proof given by Liberatore in [10] (see Lemmas 2.16 and 3.10).

Theorem 1. *If there exists an α-approximation algorithm for $(\frac{1}{3}, \frac{1}{2})$-separator and a β-approximation algorithm for DPLA, then the above algorithm achieves an approximation factor of $O(\alpha + \beta)$ for DCA.*

Proof. First, we show that arranging the edges of the approximate separator of G in the final circular arrangement, costs no more than $O(\alpha) \cdot OPT_{DCA}(G)$.

Let f^* be an optimal directed circular arrangement. Let us look at the following partitions for each value of $i \in \{0, 1, 2, \ldots, n-1\}$:

$$S_1(i) = \Big\{f^{*-1}\big(j \mod n\big) | i+1 \leq j \leq i + \frac{n}{3}\Big\}$$
$$S_2(i) = \Big\{f^{*-1}\big(j \mod n\big) | i+\frac{n}{3}+1 \leq j \leq i + \frac{2n}{3}\Big\}$$
$$S_3(i) = \Big\{f^{*-1}\big(j \mod n\big) | i+\frac{2n}{3}+1 \leq j \leq i + n\Big\}$$

One can view an example in Figure 3.

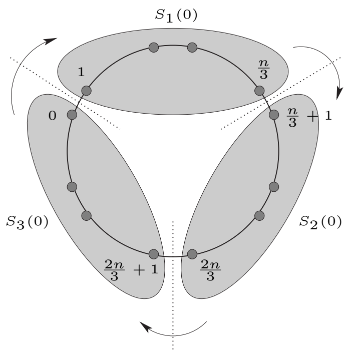

Figure 3: $S_1(0)$, $S_2(0)$, $S_3(0)$ and the edges that have endpoints in different sets

The goal is that for each possible value of i, all sets form a legal partition of the vertex set V and each set in the partition has at most $\frac{n}{3}$ vertices. Assume without loss of generality that n is a multiple of 3 (we note that for simplicity of the proof we do not consider the case where n is not a multiple of 3, although this case can also be handled).

For each value of $i \in \{0, 1, 2, \ldots, n-1\}$, the total weight of the edges that have two endpoints in different sets is bounded from below by the weight of the optimal $\frac{1}{3}$-separator. By summing up over all possible values of i, and since the weight of each edge contributes to the sum at most three times the length of the edge in f^*, we obtain that the total weight of such edges is bounded from above by $3 \cdot OPT_{DCA}(G)$. Thus, we can conclude the following:

$$n \cdot OPT_{\frac{1}{3}\text{-separator}}(G) \leq 3 \cdot OPT_{DCA}(G).$$

The edges that are found in Step (1) of the algorithm form a legal $\frac{1}{2}$-separator and have a total weight which is not more than $\alpha \cdot OPT_{\frac{1}{3}-separator}(G)$. Since each edge has length of no more than $n-1$ in *any* circular arrangement, arranging these edges in the directed circular arrangement created by **APPROXDCA** costs no more than $n-1$ times their weight. Therefore, the total cost of arranging the edges removed in Step (1) of the algorithm is at most:

$$3\alpha \cdot OPT_{DCA}(G) = O(\alpha) \cdot OPT_{DCA}(G).$$

Consider the graph: $\cup_{i=1}^{k} G_i$. In [10] it was proved that there exists an optimal directed circular arrangement which places the vertices of G_i, $1 \leq i \leq k$, next to each other (refer to section 3 lemma 11 in [10]). Let us denote such an arrangement by f^{**}. The cost of the component G_i in f^{**}, i.e., the cost of the edges in E_i in the arrangement f^{**}, is bounded from below by $OPT_{DPLA}(G_i)$ with penalties $q_i = 1$ and $p_i = |V \setminus V_i|$. Hence, by summing up over all components in the arrangement f^{**} we obtain that:

$$\sum_{i=1}^{k} \Big(OPT_{DPLA}(G_i) \Big) \leq OPT_{DCA}\big(\cup_{i=1}^{k} G_i \big),$$

where the penalties are $q_i = 1$ and $p_i = |V \setminus V_i|$, $1 \leq i \leq k$.

We obtain for each component a directed penalized linear arrangement that costs at most $\beta \cdot OPT_{DPLA}(G_i)$. Notice that a *penalized* edge in such an arrangement contributes at least $|V \setminus V_i| \geq \frac{1}{2}n$ times its weight. In the final circular arrangement, such an edge will have length of at most $n - 1$. A *non-penalized* edge will have the same length in the final directed circular arrangement as in the directed penalized linear arrangement. Thus, we get that arranging each component according to the order obtained via the approximate directed penalized linear arrangement will cost us no more than twice the value of the approximate penalized arrangement of that component. Therefore, we can conclude that arranging all the components, as described in the algorithm, costs at most:

$$2 \sum_{i=1}^{k} \Big(\beta \cdot OPT_{DPLA}(G_i) \Big) \leq O(\beta) \cdot OPT_{DCA}\big(\cup_{i=1}^{k} G_i \big).$$

However, since $OPT_{DCA}\big(\cup_{i=1}^{k} G_i \big) \leq OPT_{DCA}(G)$, we can conclude that the cost of arranging all k components is at most $O(\beta) \cdot OPT_{DCA(G)}$. Hence, the total cost of the directed circular arrangement found by **APPROXDCA** is at most: $O(\alpha + \beta) \cdot OPT_{DCA}(G)$. □

By applying the best known approximation factor for separators ([5]) together with Theorem 9, we obtain an $O(\log n \log \log n)$ approximation factor for DCA. *

5 Directed Penalized Linear Arrangement

In this section we provide an $O(\log n \log \log n)$-approximation algorithm for the (p, q)-directed penalized linear arrangement (DPLA) problem. We can assume without loss of generality that the graph G contains all possible directed edges $(u, v) \in V$, since we can add all missing edges to the graph and set their weight to 0.

We use region growing techniques to approximate the DPLA problem by defining two distinct sets of directed metrics. In addition, we define a sequence of new directed spreading metrics that are used for applying the algorithm recursively on smaller subgraphs. By *inflating* distances and *deflating* weights, we use the new spreading metrics to define an asymmetric region growing procedure that accounts simultaneously for both incoming and outgoing edges.

The complexity of our algorithm, in comparison with other divide and conquer algorithms for similar arrangement problems, e.g., linear arrangement, is due to the fact that the directed

*Note that instead of using (ρ, ρ')-separators, we can use *balanced cuts*. The latter problem has an improved approximation factor of $O(\sqrt{\log n})$ due to [1]. Our approximation factor is determined by the Theorem 9 which gives a larger factor of $O(\log n \log \log n)$, thus making the use of separators and balanced cuts equivalent.

penalized linear arrangement problem is highly asymmetric. In this problem, there are two types of edges, *penalized* and *non-penalized*. Each type of edge has a different contribution to the objective function. A penalized edge contributes to the objective function a term which is only proportional to its weight and it is independent of its length in the arrangement. On the other hand, a non-penalized edge contributes to the objective function a term which is proportional to its weighted length in the arrangement. The need to distinguish between these cases makes our algorithm more complex than usual.

5.1 A Directed Double Spreading Metric.

We define a directed *double* spreading metric. The term double is used since each edge e is associated with two length functions: the *right* length, $d_R(e)$, and the *left* length, $d_L(e)$. Right lengths correspond to non-penalized lengths and left lengths correspond to penalized lengths. Using the right and left length functions we define a sequence of directed metrics, $\{d_k\}_{k=2}^n$, such that directed metric d_k is intended to be used on subgraphs having k vertices.

$$
\begin{aligned}
\min \quad & \sum_{u,v \in V} w(u,v)\big(q d_R(u,v) + p d_L(u,v)\big) & & \\
s.t. \quad & d_k(u,v) \le d_R(u,v) & \forall u,v \in V, \forall k \in \{2,3,\ldots,n\} \quad & (1) \\
& d_k(u,v) \le k d_L(v,u) & \forall u,v \in V, \forall k \in \{2,3,\ldots,n\} \quad & (2) \\
& d_k(u,v) + d_k(v,w) \ge d_k(u,w) & \forall u,v,w \in V, \forall k \in \{2,3,\ldots,n\} \quad & (3) \\
& \sum_{v \in S} \big(d_k(u,v) + d_k(v,u)\big) \ge \frac{1}{4}\big(k^2 - 1\big) & \forall S \subseteq V, |S| = k, \forall u \in S, \forall k \in \{2,3,\ldots,n\} \quad & (4) \\
& d_L(u,v), d_R(u,v), d_k(u,v) \ge 0 & \forall u,v \in V, \forall k \in \{2,3,\ldots,n\} \quad & (5)
\end{aligned}
$$

A directed double spreading metric is an optimal solution to the above linear program. We denote the value of an optimal solution by τ.

Lemma 2. *τ is a lower bound on the value of an optimal solution to the DPLA problem.*

Proof. We show that any solution f to the DPLA problem defines a feasible solution to the above linear program, such that the objective function value of the linear program is equal to the value of the solution f. For each edge $(u \to v) \in E$:

- if $f(u) < f(v)$, then set $d_R(u,v) = f(v) - f(u)$ and $d_L(u,v) = 0$;
- otherwise, if $f(u) > f(v)$, then set $d_R(u,v) = 0$ and $d_L(u,v) = 1$.

For all k and for all $u, v \in V$ we set: $d_k(u,v) = \min\{k \cdot d_L(v,u), d_R(u,v)\}$. Thus, for each edge $(u \to v) \in E$ that goes to the right in f, $d_k(u,v) = \min\{k, d_R(u,v)\}$, and for each edge $(u \to v) \in E$ that goes to the left in f, $d_k(u,v) = 0$ (since $d_R(u,v) = 0$ and $d_L(v,u) = 0$). One can view these assignments in Figure 4.

Clearly, this assignment satisfies constraints (1), (2), and (5), and the distances d_k, $k \in \{2,3,\ldots,n\}$, satisfy the triangle inequality (constraint (3)). Therefore, d_k is a directed metric.

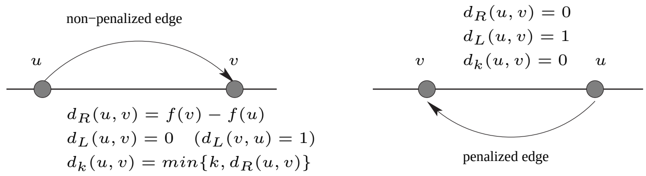

Figure 4: Assignments to d_R, d_L and d_k.

All that remains for us to prove is that constraint (4), the spreading constraint, is satisfied. Notice that for every k and for every u and v: $d_k(u,v) + d_k(v,u) = \min\{k, |f(v) - f(u)|\}$. Therefore, one can view $d_k(u,v) + d_k(v,u)$ as the truncated undirected distance between u and v in the arrangement f. For any subset $S \subseteq V$ of cardinality k and a vertex $u \in S$, the sum of the truncated undirected distances between u and the rest of the vertices in S is minimized when the rest of the $k-1$ vertices are evenly "packed" around u. Therefore,

$$\sum_{v \in S} \big(d_k(u,v) + d_k(v,u)\big) \geq \sum_{i=1}^{\lceil \frac{k-1}{2} \rceil} i + \sum_{i=1}^{\lfloor \frac{k-1}{2} \rfloor} i \geq \frac{1}{4}\big(k^2 - 1\big).$$

We conclude that the defined solution for the linear program is feasible.

We now examine the contribution of each edge $(u \to v) \in E$ to the objective function value. If $f(u) < f(v)$, then $d_R(u,v) = f(v) - f(u)$ and $d_L(u,v) = 0$. Thus, the contribution of an edge that goes to the right is $w(u,v)(qd_R(u,v) + pd_L(u,v)) = w(u,v)q(f(v) - f(u))$. Otherwise, if $f(u) > f(v)$, then $d_R(u,v) = 0$ and $d_L(u,v) = 1$. Therefore, the contribution of an edge that goes to the left is $w(u,v)(qd_R(u,v) + pd_L(u,v)) = p \cdot w(u,v)$. Hence, the value of the solution is exactly the value of f, yielding that τ is a lower bound on the value of an optimal solution to the DPLA problem. □

We note that in the above proof both d_R and d_L induce a directed metric. The metric d_R is similar to the metric used for the DLA problem. The metric d_L is similar to the metric used for the directed feedback edge set problem [12], since the directed feedback edge set problem can be formulated as an arrangement problem, where the feedback edges are, say, the edges going to the "left" (and whose length is independent of the arrangement).

Lemma 3. *A directed double spreading metric is computable in polynomial time.*

Proof. The number of variables in the linear program is polynomial. There is a polynomial number of constraints of type (1), (2), (3) and (5). Thus, we only need to present a separation oracle for the spreading constraints, (4), since there is an exponential number of them. Then we can use the ellipsoid algorithm to compute an optimal solution in polynomial time.

We show how a separation oracle can check in polynomial time if all constraints of this type are satisfied, and if not, return an unsatisfied constraint. For each value of k ($k \in \{2, \ldots, n\}$) and for each vertex $u \in V$, we can find the $k-1$ closest vertices to u by sorting all vertices $v \in V$ ($v \neq u$) with respect to $\ell_k(v) = d_k(u,v) + d_k(v,u)$. Clearly, if constraint (4) holds for the $k-1$ closest vertices to u, then it holds for any other set S for which $|S| = k$ and $u \in S$. Thus, we have a polynomial time separation oracle. □

5.2 Asymmetric Region Growing

5.2.1 The Volume Function.

A sphere is associated with a *root* vertex s and radius r. Define

$$N_k(s,r) = \{v | d_k(s,v) < r\}.$$

Let us define the set of edges that belong to a sphere as

$$E_k(s,r) = E \cap (N_k(s,r) \times N_k(s,r)).$$

Since the graph is directed, there is a need to distinguish between edges that go *into* the sphere and edges that go *out* of the sphere. This brings us to the following definitions of the *right* edges in the cut induced by a sphere (edges that go to the right in the final arrangement and are *non-penalized*), and *left* edges (edges that go to the left in the final arrangement and are *penalized*). We use sub-index R for the right edges and sub-index L for the left edges. Thus,

$$Cut_{k,R}(s,r) = \{(u \to v) \in E | u \in N_k(s,r), v \notin N_k(s,r)\}$$
$$Cut_{k,L}(s,r) = \{(u \to v) \in E | u \notin N_k(s,r), v \in N_k(s,r)\}.$$

The *volume* of a sphere is intuitively the sum of the weighted lengths of the edges that are contained inside the sphere, plus a fraction of the weighted lengths of edges that have precisely one vertex outside the sphere. (The fraction corresponds to the part of the edge that is contained inside the sphere.) However, due to the need to distinguish between *penalized* and *non-penalized* edges, right and left edges in the cut cannot be treated in the same way.

Assume that we partition a graph with k vertices according to a cut induced by a sphere rooted at s with radius r. We place the vertices of the sphere on the left in the arrangement. Edges that become non-penalized (edges that go out of the sphere and belong to $Cut_{k,R}(s,r)$) contribute to the value of the resulting arrangement at most $p(k-1)$ times their weight. Edges that become penalized (edges that go into the sphere and belong to $Cut_{k,L}(s,r)$) contribute to the value of the resulting arrangement exactly q times their weight.

Non-penalized edges are treated similarly to [4] and [5]. However, the treatment of penalized edges is different. Since $d_k(u,v)$ is bounded from above by the "inflated" left length of the opposite edge, $kd_L(v,u)$ (the inflation is by a factor of k, see constraint (2) in the double spreading metric), we are about to "deflate" the penalized weight of all left edges in the cut by the same factor of k. This explains why ingoing and outgoing edges contribute differently to the volume. We now need to prove that this indeed defines a legal volume function, where by legal we mean that the defined function has all the "nice" properties one can expect from a standard volume function. This brings us to the following definition of volume:

$$\begin{aligned} Vol_{k,s}(r) = \frac{\tau}{n} + \big(|N_k(s,r)| - 1\big)^{+}\frac{\tau}{n} + \\ \sum_{(u\to v)\in E_k(s,r)} w(u,v)\big(qd_R(u,v) + pd_L(u,v)\big) + \\ \sum_{(u\to v)\in Cut_{k,R}(s,r)} q\cdot w(u,v)\big(r - d_k(s,u)\big) + \\ \sum_{(u\to v)\in Cut_{k,L}(s,r)} \frac{p\cdot w(u,v)}{k}\big(r - d_k(s,v)\big), \end{aligned}$$

(where $(x)^+ = x$ if $x \geq 0$ and $(x)^+ = 0$ if $x < 0$). An example can be seen in Figure 5. Thus,

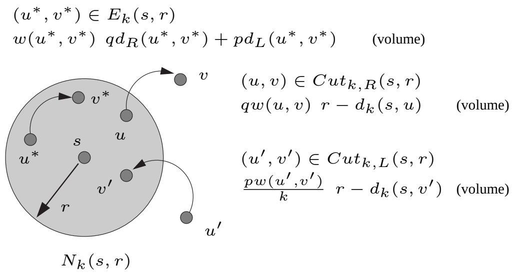

Figure 5: Different contribution of edges to $Vol_{k,s}(r)$.

intuitively, by inflating the lengths and deflating the weights by the same factor we can preserve the volume of the metric and bound the cost of *both* incoming and outgoing edges. Let us remark that this inflation enabled us to prove in Lemma 2 that d_k, which was bounded from above by the *inflated* left length of the reverse edge, i.e., $d_k(u,v) \leq kd_L(v,u)\ \forall u,v \in V$, satisfies the spreading constraints.

Note that each volume function has an additional *seed value*, which is the first line in the definition of the volume function. These seed values guarantee that for any root vertex s and any radius r, the volume functions are always strictly positive, specifically at least $\frac{\tau}{n}$. The seed values are chosen such that their contribution to the volume is bounded, otherwise the approximation guarantee might not hold. This bound is the following: $\frac{\tau}{n} + \big(|N_k(s,r)| - 1\big)^{+}\frac{\tau}{n} \leq \frac{\tau}{n} + (n-1)\frac{\tau}{n} = \tau$ (since $|N_k(s,r)| \leq n$). The reason for choosing this exact seed value will become clear when proving the approximation guarantee of the algorithm.

There is an asymmetry in the volume definitions since DPLA is defined for directed graphs. The "ordinary" region growing, as used in [4, 5, 8] for either undirected graphs or directed graphs, cannot be used here. The reason is that if spheres are grown according to distances from the root vertex, only the cost of edges that go out of the sphere can be bounded. Alternatively, if spheres are grown according to distances to the root vertex, only the cost of edges that go into the sphere can be bounded. Thus, for directed graphs, "ordinary" sphere growing can only be used when it suffices to bound the cost of either outgoing edges, or ingoing edges, but not both. This is the case, for example, in the directed feedback edge set problem [12]. In DPLA, in contrast, we need to bound the cost of both types of edges.

We now show that the volume functions have "nice" properties. These properties are formally stated in the following lemma. The motivation for requiring these properties from the volume functions is that we want to use region growing techniques similarly to [5, 8].

Lemma 4. *$Vol_{k,s}(r)$ is a positive, monotone non-decreasing function in r that is differentiable almost everywhere and its derivative is continuous almost everywhere, for every vertex s and for every k.*

Proof. Clearly, for each vertex $s \in V$ and for every k, $Vol_{k,s}(r)$ is positive, differentiable almost everywhere and its derivative is continuous almost everywhere. In order to show that $Vol_{k,s}(r)$ is a monotone, non-decreasing, function in r, we need to show the following: (i) the coefficients of r are non-negative; (ii) the contribution of each edge to $Vol_{k,s}(r)$, when the edge is in a cut, is not more than its contribution when it is in a sphere. One can see that the coefficients of r are indeed non-negative, therefore we concentrate on (ii).

Let $e = (u \to v) \in E$ be an edge. The contribution of the edge e to $Vol_{k,s}(r)$ while it belongs to the sphere is: $w(u,v)\big(qd_R(u,v) + pd_L(u,v)\big)$. If $e \in Cut_{k,R}(s,r)$, then $d_k(s,u) < r$ and $d_k(s,v) \geq r$. Thus, we get:

$$\begin{aligned} q \cdot w(u,v)(r - d_k(s,u)) &\leq q \cdot w(u,v)\big(d_k(s,v) - d_k(s,u)\big) \\ &\leq q \cdot w(u,v) d_k(u,v) \\ &\leq q \cdot w(u,v) d_R(u,v) \\ &\leq w(u,v)\big(qd_R(u,v) + pd_L(u,v)\big). \end{aligned}$$

The second inequality is derived from the directed triangle inequality (constraint (3) of the double spreading metric). The third inequality is derived from constraint (1) of the double spreading metric. If $e \in Cut_{k,L}(s,r)$, then $d_k(s,v) < r$ and $d_k(s,u) \geq r$. Hence, we get:

$$\begin{aligned} \frac{p \cdot w(u,v)}{k}\big(r - d_k(s,v)\big) &\leq \frac{p \cdot w(u,v)}{k}\big(d_k(s,u) - d_k(s,v)\big) \\ &\leq \frac{p \cdot w(u,v)}{k} d_k(v,u) \\ &\leq \frac{p \cdot w(u,v)}{k} k d_L(u,v) \\ &= p \cdot w(u,v) d_L(u,v) \\ &\leq w(u,v)\big(qd_R(u,v) + pd_L(u,v)\big). \end{aligned}$$

The second inequality is derived from the directed triangle inequality (constraint (3) of the double spreading metric). The third inequality is derived from constraint (2) of the double spreading metric. Therefore, $Vol_{k,s}(r)$ is a monotone non-decreasing function in r for every vertex $s \in V$ and for every $k \in \{2, 3, \ldots, n\}$. □

5.2.2 Choosing a Cut

We now prove the existence of a "good" radius, via the following well known claim.

Lemma 5. *Let $f : [r_0, r_1] \to R^+$ be a positive, monotone non-decreasing function that is differentiable almost everywhere. If the derivative f' is continuous almost everywhere, there exists an $r \in (r_0, r_1)$ such that:*

$$f'(r) \le \frac{f(r)}{r_1 - r_0} \ln\left(\frac{e \cdot f(r_1)}{f(r)}\right) \ln\ln\left(\frac{e \cdot f(r_1)}{f(r_0)}\right).$$

Proof. Let us assume the contrary, i.e., that for every $r \in (r_0, r_1)$ (where $f'(r)$ is defined):

$$f'(r) > \frac{f(r)}{r_1 - r_0} \ln\left(\frac{e \cdot f(r_1)}{f(r)}\right) \ln\ln\left(\frac{e \cdot f(r_1)}{f(r_0)}\right).$$

Since f is positive and monotone non-decreasing, this is equivalent to:

$$\frac{f'(r)}{f(r) \ln\left(\frac{e \cdot f(r_1)}{f(r)}\right)} > \frac{1}{r_1 - r_0} \ln\ln\left(\frac{e \cdot f(r_1)}{f(r_0)}\right).$$

Since f is differentiable almost everywhere and its derivative is continuous almost everywhere, we can take the integral on both sides, resulting in:

$$\int_{r_0}^{r_1} \frac{f'(r)}{f(r) \ln\left(\frac{e \cdot f(r_1)}{f(r)}\right)} dr > \int_{r_0}^{r_1} \frac{1}{r_1 - r_0} \ln\ln\left(\frac{e \cdot f(r_1)}{f(r_0)}\right) dr.$$

Since f is monotone non decreasing, we obtain for the left-hand side that:

$$\int_{r_0}^{r_1} \frac{f'(r)}{f(r) \ln\left(\frac{e \cdot f(r_1)}{f(r)}\right)} dr \le -\ln\ln\left(\frac{e \cdot f(r_1)}{f(r_1)}\right) + \ln\ln\left(\frac{e \cdot f(r_1)}{f(r_0)}\right) = \ln\ln\left(\frac{e \cdot f(r_1)}{f(r_0)}\right).$$

By evaluating the right-hand side we obtain:

$$\int_{r_0}^{r_1} \frac{1}{r_1 - r_0} \ln\ln\left(\frac{e \cdot f(r_1)}{f(r_0)}\right) dr = \ln\ln\left(\frac{e \cdot f(r_1)}{f(r_0)}\right).$$

This is a contradiction, thus the claim follows. □

Let us now describe the procedure for region growing. Denote by $Vol'_{k,v}(r)$ the (left) derivative of $Vol_{k,v}(r)$ with respect to r.

SphereGrow($G' = (V', E')$,v,d_k)

1. $S \leftarrow \{v\}$, $i \leftarrow 0$.
2. Sort V' by $d_k(v, x)$ ($\forall x \in V'$) according to increasing order.
   Let $x_0, x_1, x_2, \ldots, x_{k-1}$ be the resulting order.
3. $\Delta \leftarrow d_k(v, x_{k-1})$.
4. While $Vol'_{k,v}\big(d_k(v, x_i)\big) > \Big[\frac{2}{\Delta} \ln\Big(\frac{eVol_{k,v}(\Delta)}{Vol_{k,v}\big(d_k(v,x_i)\big)}\Big) \ln\ln\Big(\frac{eVol_{k,v}(\Delta)}{Vol_{k,v}(0)}\Big) Vol_{k,v}\big(d_k(v, x_i)\big)\Big]$, do:
   - $S \leftarrow S \cup \{x_{i+1}\}$.
   - $i \leftarrow i + 1$.
5. Return $(S, \min\{\Delta/2, d_k(v, x_i)\})$.

The input to the above procedure consists of an input graph G' that contains k vertices, a root vertex v of the sphere, a metric d_k with respect to which the sphere is grown, and a parameter Δ such that the maximum radius of the grown sphere is at most $\Delta/2$.

The region growing procedure we use is the discrete version of the intuitive continuous region growing. We will prove that a good cut is indeed computed by the above procedure, however, we first need to define the *adjusted weight* of a cut. Assume we are partitioning a graph with k vertices into two subgraphs by a good cut. Then, the cost of edges that become non-penalized in the arrangement is at most $p(k-1)$ times their weight, and the cost of edges that become penalized is exactly q times their weight. Therefore, if we want the arrangement cost of the edges to be at most k times the weight of the cut, we need to adjust the weight of the edges that become penalized. By adjusting we mean decreasing their weight by a factor of k. This is a crucial point in the analysis.

Definition 2. *The adjusted weight of a cut $(S, V \setminus S)$ of a directed graph $G = (V, E)$, where $|V| = k$, is*

$$\sum_{(u \to v) \in E | u \in S, v \notin S} q \cdot w(u, v) + \sum_{(u \to v) \in E | u \notin S, v \in S} \frac{p \cdot w(u, v)}{k}.$$

Thus, by using the adjusted weight of a cut, we get that the contribution of the edges in the cut to the cost of the arrangement is bounded by at most k times the adjusted weight of the cut, as is evident in Figure 6. Notice that the (left) derivative of the volume function equals the adjusted weight of the cut induced by the appropriate sphere. More precisely,

$$Vol'_{k,s}(r) = \sum_{(u \to v) \in Cut_{k,R}(s,r)} q \cdot w(u, v) + \sum_{(u \to v) \in Cut_{k,L}(s,r)} \frac{p \cdot w(u, v)}{k}.$$

Th motivation for considering cuts (as in **SphereGrow**), is that the cuts are used to find an ordering of the graph via divide and conquer techniques.

Lemma 6. *We are given a subgraph $G' = (V', E')$ of G , where $|V'| = k$, a vertex $v \in V'$, and a directed metric d_k on G'. Let $x_0, x_1, \ldots, x_{k-1}$ be the sorted vertices and set $\Delta = d_k(v, x_{k-1})$*

as in **SphereGrow** *on the input* (G', v, d_k)*. Then,* **SphereGrow(**G'**,**v**,**d_k**)** *finds a non-trivial cut* $(S, V' \setminus S)$ *of* G'*, where* $S = \{x_0, x_1, \ldots, x_{m-1}\}$ *(for some* $m = 1, \ldots, k-1$*) and for every* $x \in S$*:* $d_k(v, x) < \frac{\Delta}{2}$*, whose total adjusted weight is at most:*

$$\Big[\frac{2}{\Delta} \ln\left(\frac{eVol_{k,v}(\Delta)}{Vol_{k,v}\big(\min\{\Delta/2, d_k(v, x_m)\}\big)}\right) \ln\ln\left(\frac{eVol_{k,v}(\Delta)}{Vol_{k,v}(0)}\right) Vol_{k,v}\big(\min\{\Delta/2, d_k(v, x_m)\}\big)\Big].$$

Proof. The function $Vol_{k,v}(r)$ is a piecewise linear function, where the endpoints of each linear segment correspond to $d_k(v, x)$ for some vertex $x \in V'$. Thus, for each $i \in \{0, 1, \ldots, k-2\}$, $Vol_{k,v}(r)$ is linear in the interval $\big(d_k(v, x_i), d_k(v, x_{i+1})\big)$ and its derivative in it is exactly the adjusted weight of the cut $\big(\{x_0, x_1, \ldots, x_i\}, V' \setminus \{x_0, x_1, \ldots, x_i\}\big)$. This can be easily seen by the fact that the coefficients of r correspond exactly to the adjusted weights of the edges in the cut.

By Lemma 4, the function $Vol_{k,v}(r)$ over the region $[0, \Delta]$, satisfies all the needed conditions of Lemma 5. In particular, this is also correct for the region $[0, \frac{\Delta}{2}]$. Thus, by Lemma 5, we know that there exists a radius $r^* \in (0, \frac{\Delta}{2})$, such that

$$Vol'_{k,v}(r^*) \leq \frac{2}{\Delta} \ln\left(\frac{e \cdot Vol_{k,v}(\frac{\Delta}{2})}{Vol_{k,v}(r^*)}\right) \ln\ln\left(\frac{e \cdot Vol_{k,v}(\frac{\Delta}{2})}{Vol_{k,v}(0)}\right) Vol_{k,v}(r^*).$$

According to Lemma 4, $Vol_{k,v}(r)$ is a monotone non-decreasing function. Hence, we can conclude that:

$$Vol'_{k,v}(r^*) \leq \frac{2}{\Delta} \ln\left(\frac{e \cdot Vol_{k,v}(\Delta)}{Vol_{k,v}(r^*)}\right) \ln\ln\left(\frac{e \cdot Vol_{k,v}(\Delta)}{Vol_{k,v}(0)}\right) Vol_{k,v}(r^*).$$

Therefore, all that is left to prove is that among the distances $\{d_k(v, x_i) | 0 \leq i \leq k-1, d_k(v, x_i) \leq \frac{\Delta}{2}\}$, there exists a *good* radius (i.e. a radius whose cut upholds the conditions in the lemma). Note that this shows that the procedure terminates and finds a non-trivial cut, since $d_k(v, x_{k-1}) = \Delta$. Let us define:

$$r^{**} = \min\{d_k(v, x) | x \in V', d_k(v, x) \geq r^*\}.$$

Note that r^{**} is well defined since $r^* \leq \frac{\Delta}{2}$ and $d_k(v, x_{k-1}) = \Delta$. Since $N_k(v, r^{**})$ consists of vertices whose distance from v is *strictly* less than r^{**}, we get, according to the definition of r^{**} that: $N_k(v, r^*) = N_k(v, r^{**})$. Thus, both spheres (defined by r^* and r^{**}) are the same, which implies that they both define the same cuts. Hence, we can conclude that $Vol'_{k,v}(r^{**}) = Vol'_{k,v}(r^*)$. Additionally, since $N_k(v, r^*) = \{x | d_k(v, x) < r^*\}$ and $r^* \leq \frac{\Delta}{2}$, we can conclude that each $x \in N_k(v, r^{**})$ satisfies the condition: $d_k(v, x) < \frac{\Delta}{2}$. According to Lemma 4, $Vol_{k,v}(r)$ is a monotone non-decreasing function. Since the function $x \ln \frac{ea}{x}$ is monotone non-decreasing in the interval $(0, a]$, we can conclude that:

$$\begin{aligned} Vol'_{k,v}(r^{**}) = Vol'_{k,v}(r^*) &\leq \frac{2}{\Delta} \ln\left(\frac{e \cdot Vol_{k,v}(\Delta)}{Vol_{k,v}(r^*)}\right) \ln\ln\left(\frac{e \cdot Vol_{k,v}(\Delta)}{Vol_{k,v}(0)}\right) Vol_{k,v}(r^*) \\ &\leq \frac{2}{\Delta} \ln\left(\frac{e \cdot Vol_{k,v}(\Delta)}{Vol_{k,v}(r^{**})}\right) \ln\ln\left(\frac{e \cdot Vol_{k,v}(\Delta)}{Vol_{k,v}(0)}\right) Vol_{k,v}(r^{**}). \end{aligned}$$

Therefore, r^{**} is a *good* radius. Note that one can conclude from the above that $d_k(v, x_m) = r^{**}$.

The only problem that might arise is that $r^{**} > \Delta/2$. In this case since $r^* \leq \Delta/2 \leq r^{**}$, we can conclude that $N_k(v, r^{**}) = N_k(v, \Delta/2)$. Then by increasing the radius r^* to $\Delta/2$ instead of r^{**}, we get the needed result. This finishes the proof. □

5.3 An Approximation Algorithm for Directed Penalized Linear Arrangement.

Before stating the approximation algorithm, we need the following definition which will help us establish a connection between the directed metrics $\{d_k\}_{k=2}^n$, used in the region growing procedure, and the distances d_R and d_L which form the lower bound on the optimum, τ.

Definition 3. *The full volume of a non-empty subgraph $G' = (V', E')$ of G will be denoted by $Vol(G')$ and is equal to:*

$$\sum_{(u,v)\in E'} w(u,v)\big(qd_R(u,v) + pd_L(u,v)\big) + |V'| \cdot \frac{\tau}{n}.$$

Note that according to this definition $Vol(G) = 2\tau$. Let us now state the approximation algorithm for directed penalized linear arrangement. We assume that the directed double spreading metric has been found for the input graph, and use the same metric recursively.[†]

APPROXDPLA($G = (V, E)$)

1. Let $k \leftarrow |V|$, $(s,t) = \text{argmax}_{(s,t)\in V\times V}\{d_k(s,t)\}$.
2. $(S, r) \leftarrow$ **SphereGrow(G,s,d_k)**.
   Assume without loss of generality that $Vol_{k,s}(r) \leq \frac{Vol(G)}{2}$.
3. Place in the arrangement subgraph G_1, induced by S, on the left,
   and subgraph G_2, induced by $V \setminus S$, on the right.
4. **APPROXDPLA(G_1)**, **APPROXDPLA(G_2)**.

The partitioning step is shown in Figure 6.

We show in the following lemma that the spreading constraint gives us a directed radius guarantee. The need for this radius guarantee is that according to Lemma 6, the adjusted weight of the cut found by the algorithm is inversely dependent on the radius of the graph. The bigger the radius, the smaller the *adjusted weight* of the cut found by the algorithm. Additionally, the cost of arranging the edges in a cut is at most k times the *adjusted weight* of the cut (where k is the number of vertices in the subgraph). Therefore, if we can show that in each subgraph containing k vertices we have a directed radius guarantee of $\Omega(k)$, then we can successfully bound the cost of the solution found by the algorithm (see Lemma 8 and Theorem 9). By directed radius guarantee of $\Omega(k)$ we mean that there is some pair of vertices whose distance is at least $\Omega(k)$. This is stated in the following lemma:

Lemma 7. *For every subset $S \subseteq V$ such that $|S| = k$ ($k \geq 2$) and for every vertex $u \in S$, there exists a vertex v such that: $d_k(u,v) = \Omega(k)$ or $d_k(v,u) = \Omega(k)$.*

[†]In the algorithm we assume without loss of generality that $Vol_{k,s}(r) \leq \frac{Vol(G)}{2}$. This assumption can be made since one can define a reverse volume function and a reverse region growing procedure which are similar to the regular definitions, except for the fact that they use distances *to* the root of the sphere instead of distances *from* the root of the sphere. It can be proved that the sum of volumes of the regular sphere rooted at s and the reverse sphere rooted at t (where s and t are chosen such that $(s,t) = \text{argmax}_{(s,t)\in V\times V}\{d_k(s,t)\}$) is at most $Vol(G)$. Therefore, by choosing the sphere with the less volume, one finds a sphere whose volume is at most $\frac{Vol(G)}{2}$.

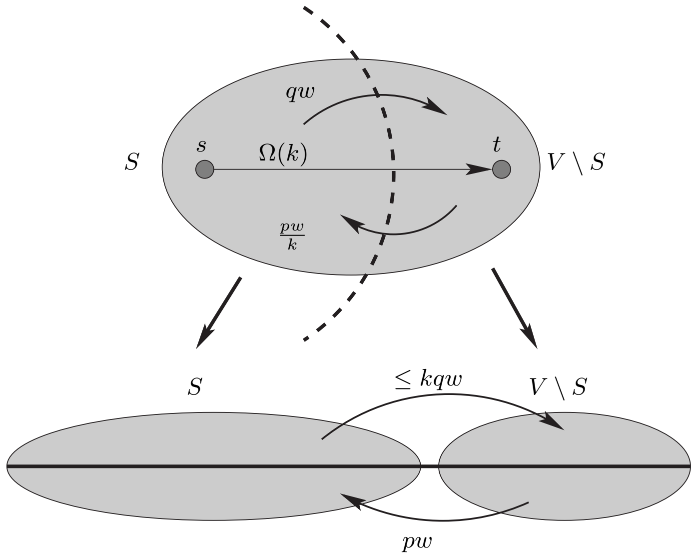

Figure 6: The partitioning step in **APPROXDPLA**. Note the adjusted cut weight and the cost of the edges in the arrangement. The weight of the edge is w and k is the number of vertices in the current graph.

Proof. By the spreading constraint for S and u we get that: $\sum_{v\in S}\big(d_k(u,v)+d_k(v,u)\big) \geq \frac{1}{4}(k^2-1)$. Since $|S| = k$, there exists a vertex $v \in S$ such that $d_k(u,v)+d_k(v,u) = \Omega(k)$. By choosing the larger between $d_k(u,v)$ and $d_k(v,u)$ we finish the proof. □

We establish the approximation factor of the algorithm via the following charging scheme. Let $g(\eta)$ denote the maximum arrangement cost computed by **APPROXDPLA** of a subgraph of G having a full volume of at most η. Note that according to this definition the function g is monotone non-decreasing. Clearly, by the full volume definition, we get that $g(\eta) = 0$ for every $0 \leq \eta < \frac{\tau}{n}$, since we are left with no vertices.

Let us now examine what happens when **APPROXDPLA** is invoked on a subgraph G' of G induced by $V' \subseteq V$, where $|V'| = k$. Let us denote by η the full volume of G'. Let us also denote by η_1 the volume of the sphere currently chosen by **APPROXDPLA**, i.e., $Vol_{k,s}(r)$. According to our assumption, $\eta_1 \leq \frac{\eta}{2}$.

As stated before, a crucial point in the analysis is that the arrangement cost of edges defined by a cut in a graph with k vertices, is at most k times the *adjusted weight* of the cut. This is true since each edge that is non-penalized, contributes to the adjusted weight of the cut q times its weight. However, edges that are penalized, contribute to the cut p times their weight divided by k. Non-penalized edges cost in the arrangement at most $q(k-1)$ times their weight, and penalized edges cost in the arrangement p times their weight. Therefore, the arrangement cost of edges defined by a cut is at most k times the *adjusted weight* of the cut.

Hence, by Lemmas 6 and 7, we conclude that the arrangement cost of the edges in the cut chosen by **APPROXDPLA** when invoked on G' is at most:

$$c \cdot \eta_1 \ln\Big(\frac{e\cdot\eta}{\eta_1}\Big) \ln\ln\Big(2e\cdot n\Big),$$

since $\Delta = \Omega(k)$, $\eta = Vol_{k,v}(\Delta)$, $\eta_1 = Vol_{k,v}\big(\min\{\Delta/2, d_k(v, x_m)\}\big)$ and $\frac{Vol_{k,v}(\Delta)}{Vol_{k,v}(0)} \leq \frac{2\tau}{\tau/n} = 2n$ (c is an absolute constant).

In addition to arranging the edges in the cut, **APPROXDPLA** continues recursively on each of the two components left. The cost of arranging the subgraph induced by S is at most $g(\eta_1)$, since g is monotone non-decreasing. Similarly, the cost of arranging the graph induced by $V' \setminus S$ is at most $g(\eta - \eta_1)$. By summing over all costs, we get the following recursive formula:

$$g(\eta) \leq g(\eta_1) + g(\eta - \eta_1) + c \cdot \eta_1 \ln\Big(\frac{e \cdot \eta}{\eta_1}\Big) \ln\ln\Big(2e \cdot n\Big),$$

where $\frac{\tau}{n} \leq \eta_1 \leq \frac{1}{2}\eta$. Equivalently, the recursion can be written as follows:

$$g(x + y) \leq g(x) + g(y) + c \cdot x \ln\Big(\frac{e(x + y)}{x}\Big) \ln\ln\Big(2e \cdot n\Big),$$

where $\frac{\tau}{n} \leq x \leq y$.

Let us define the function $F : [\frac{\tau}{n}, \infty) \to R^+$ as follows:

$$F(x) = 3c \cdot x \ln\Big(\frac{nx}{\tau}\Big) \ln\ln\Big(2e \cdot n\Big).$$

The next lemma states that F is an upper bound on any solution of the above recursion.
Lemma 8. *F is an upper bound on the solution to the above recursion.*

Proof. It is sufficient to show that:

$$F(x + y) - F(x) - F(y) \geq c \cdot x \ln\Big(\frac{e(x + y)}{x}\Big) \ln\ln\Big(2e \cdot n\Big).$$

By calculating directly we get:

$$\begin{aligned}
F(x + y) - F(x) - F(y) &= 3c \cdot (x + y) \ln\Big(\frac{n(x + y)}{\tau}\Big) \ln\ln\Big(2e \cdot n\Big) \\
&\quad - 3c \cdot x \ln\Big(\frac{nx}{\tau}\Big) \ln\ln\Big(2e \cdot n\Big) - 3c \cdot y \ln\Big(\frac{ny}{\tau}\Big) \ln\ln\Big(2e \cdot n\Big) \\
&= 3c \cdot x \ln\Big(\frac{x + y}{x}\Big) \ln\ln\Big(2e \cdot n\Big) + 3c \cdot y \ln\Big(\frac{x + y}{y}\Big) \ln\ln\Big(2e \cdot n\Big) \\
&\geq 3c \cdot x \ln\Big(\frac{x + y}{x}\Big) \ln\ln\Big(2e \cdot n\Big).
\end{aligned}$$

However, since $x \leq y$ we get that $\frac{x+y}{x} \geq \sqrt{e}$, which implies that:

$$\ln\Big(\frac{x + y}{x}\Big) \geq \frac{1}{2}.$$

Hence, we can conclude:

$$\begin{aligned}
&3c \cdot x \ln\Big(\frac{x+y}{x}\Big) \ln\ln\Big(2e \cdot n\Big) \\
&= 2c \cdot x \ln\Big(\frac{x+y}{x}\Big) \ln\ln\Big(2e \cdot n\Big) + c \cdot x \ln\Big(\frac{x+y}{x}\Big) \ln\ln\Big(2e \cdot n\Big) \\
&\geq 2c \cdot x \frac{1}{2} \ln\ln\Big(2e \cdot n\Big) + c \cdot x \ln\Big(\frac{x+y}{x}\Big) \ln\ln\Big(2e \cdot n\Big) \\
&= c \cdot x \ln\ln\Big(2e \cdot n\Big) + c \cdot x \ln\Big(\frac{x+y}{x}\Big) \ln\ln\Big(2e \cdot n\Big) \\
&= c \cdot x \ln\Big(\frac{e(x+y)}{x}\Big) \ln\ln\Big(2e \cdot n\Big).
\end{aligned}$$

Therefore, F is an upper bound on the solution to the above recursion. □

Theorem 9. APPROXDPLA *achieves an approximation factor of* $O(\log n \log\log n)$ *for DPLA.*

Proof. Since F is an upper bound on the solution to the above recursion, the cost of the solution **APPROXDPLA** yields for a given graph G, where τ is the value of the directed double spreading metric calculated on G, is at most:

$$F(2\tau) = 6c\tau \ln\Big(\frac{n2\tau}{\tau}\Big) \ln\ln\Big(2e \cdot n\Big) = O(\log n \log\log n)\tau.$$

By Lemma 2, $\tau \leq OPT_{DPLA}$. Hence, we get that the cost of the solution that **APPROXDPLA** returns is at most $O(\log n \log\log n) \cdot OPT_{DPLA}$. □

6 Undirected Circular Arrangement

The *undirected circular-arrangement* (UCA) problem is defined as follows. The input is an undirected graph, and the objective is to find an embedding f of the vertex set onto n evenly spaced points on a circle. The length of each edge is the shortest distance on the circle, and the objective function is minimizing the weighted sum of the lengths of the edges. That is, we need to find a one-to-one function $f : V \rightarrow \{1, 2, \ldots, n\}$ which minimizes:

$$\sum_{(u,v)\in E} w(u,v) \min\Big\{\big(f(v) - f(u)\big) \mod n, \big(f(u) - f(v)\big) \mod n\Big\}.$$

The UCA problem is NP-hard [10]. In the *undirected linear arrangement* (ULA) problem, the goal is to find an embedding f of the vertex set onto n evenly-spaced points on a line while minimizing the weighted sum of the lengths of the edges. That is, we need to find a one-to-one function $f : V \rightarrow \{1, 2, \ldots, n\}$ which minimizes:

$$\sum_{(u,v)\in E} w(u,v)\big|f(v) - f(u)\big|.$$

The best approximation factor for this problem is $O(\sqrt{\log n}\log\log n)$ obtained by [3, 7].

Liberatore [10] presented an approximation algorithm for the UCA problem using (approximate) balanced cuts and (approximate) linear arrangements. The approximation factor obtained is $O(\log n)$ (based on the results of [9, 4, 11]); in fact, the approximation factor of the algorithm of Liberatore [10] can be improved to $O(\sqrt{\log n}\log\log n)$ using the recent work of [3, 7].

We now present a very simple algorithm for the UCA problem that achieves the same approximation factor. It uses the following semi-definite spreading metric, as in [3, 7].

$$\begin{array}{lll} \min & \sum_{(u,v)\in E} w(u,v)\cdot ||u-v||_2^2 & \\ s.t. & \sum_{v\in S} ||u-v||_2^2 \geq \frac{1}{4}\left(|S|^2-1\right) & \forall S\subseteq V, \forall u\in S \\ & ||u-v||_2^2 + ||v-w||_2^2 \geq ||u-w||_2^2 & \forall u,v,w\in V \end{array}$$

The main observation is that the above is also a valid spreading metric for the UCA problem. Thus, the value of an optimal solution for the above semi-definite program, τ, satisfies $\tau \leq OPT_{UCA}(G)$. The algorithm is as follows:

APPROXUCA

1. Find an approximate undirected linear arrangement using the algorithm of [3] or [7].
2. Close the linear arrangement into a circle (i.e., connect points 1 and n).

Theorem 10. **APPROXUCA** *achieves an approximation factor of* $O(\sqrt{\log n}\log\log n)$.

Proof. Step (1) of the algorithm finds a linear arrangement that costs at most $O(\sqrt{\log n}\log\log n)\cdot\tau$. By closing this linear arrangement into a circular arrangement in Step (2), the length of the edges can only decrease. Thus, the cost of the circular arrangement that **APPROXUCA** finds is at most $O(\sqrt{\log n}\log\log n)\cdot\tau$. The theorem follows since $\tau \leq OPT_{UCA}(G)$. □

We have shown that UCA can be approximated via ULA. We now prove that a good approximation for UCA also yields a good approximation for ULA.

Theorem 11. *If there is an* α*-approximation for UCA, then there is a* 2α*-approximation for ULA.*

Proof. First, it is easy to see that for any input graph G,

$$OPT_{UCA}(G) \leq OPT_{ULA}(G).$$

An α-approximation algorithm yields an undirected circular arrangement that costs at most

$$\alpha\cdot OPT_{UCA}(G) \leq \alpha\cdot OPT_{ULA}(G).$$

If we "cut" the circular arrangement at a uniformly random point, we obtain a linear arrangement in which the expected length of each edge is at most twice its length in the circular arrangement. Hence, if we choose the cutting point to be the point that gives us the best linear arrangement, we obtain a linear arrangement that costs at most $2\alpha\cdot OPT_{ULA}(G)$. □

References

[1] S. Arora, S. Rao, and U. Vazirani. Expander flows and a $\sqrt{\log n}$-approximation to sparsest cut, *Proceedings of the 36th ACM Symposium on Theory of Computing*, pp. 222-231, 2004.

[2] A. Bar-Noy, J. Naor, and B. Schieber, *Pushing dependent data in clients-providers-servers systems*, Wireless Networks, Vol. 9, (2003) pp. 421-430.

[3] M. Charikar, M. T. Hajiaghayi, H. Karloff, and S. Rao. ℓ_2^2 *spreading metrics for vertex ordering problems*, Proc. 17th Annual ACM-SIAM symposium on Discrete algorithms, 2006, pp. 1018-1027.

[4] G. Even, J. Naor, S. Rao, and B. Schieber, *Divide and Conquer Approximation Algorithms via Spreading Metrics*, Journal of the ACM, Vol. 47 (2000), pp. 585-616.

[5] G. Even, J. Naor, S. Rao, and B. Schieber, *Fast Approximate Graph Partitioning Algorithms*, SIAM Journal on Computing, Vol.28 (1998), pp. 2187-2214.

[6] G. Even, J. Naor, B. Schieber and M. Sudan, *Approximating minimum feedback sets and multicuts in directed graphs*, Algorithmica, Vol. 20 (1998), pp. 151-174.

[7] U. Feige and J. Lee, *An improved approximation ratio for the minimum linear arrangement problem*, Information Processing Letters, Vol. 101 (2007), pp. 26-29.

[8] N. Garg, V.V. Vazirani, and M. Yannakakis, *Approximate max-flow min-(multi) cut theorems and their applications*, SIAM Journal on Computing, Vol. 25 (1996), pp. 235-251.

[9] T. Leighton and S. Rao, *Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms* Journal of the ACM, Vol. 46 (1999), pp. 787-832.

[10] V. Liberatore, *Circular Arrangements*, Proc. 29th International Colloquium on Automata, Languages and Programming, 2002, pp. 1054-1065.

[11] S. Rao and A. Richa, *New Approximation Techniques for Some Ordering Problems*, Proc. 9th Annual ACM-SIAM symposium on Discrete algorithms, 1998, pp. 211-218.

[12] P. Seymour, *Packing Directed Circuits Fractionally*, Combinatorica, Vol. 15 (1995), pp. 281-288.