On Minimum Circular Arrangement [1]

Murali K Ganapathy [a] Sachin P Lodha [b]

[a] *Dept of Computer Science, University of Chicago, Chicago, Illinois, USA*

[b] *Tata Research Development and Design Centre, Pune, Maharashtra, India*

Abstract

Motivated by a scheduling problem encountered in multicast environments, we study a vertex labelling problem, called Directed Circular Arrangement (DCA), that requires one to find an embedding of a given weighted directed graph into a discrete circle which would minimize the total weighted arc length. Its decision version is already known to be NP-complete when restricted to sparse weighted directed graphs. We prove that the decision version of even un-weighted DCA is NP-complete in case of arbitrary given densities.

We also consider complementary version of DCA, called MaxCA. We prove that it is MAX-SNP[π] complete and, therefore, has no PTAS unless P=NP. A similar proof technique shows that DCA is MAX-SNP[π]-hard and hence there is no PTAS for DCA as well.

Key words: Computational complexity; hardness of approximation; polynomial time approximation scheme; scheduling; multicast

1 Introduction

Availability of very high-speed and large bandwidth networks, explosion in inter-networking, and advent of cheap, low-power, portable computing devices have given rise to one-to-many *asymmetric* communication networks and huge client populations having commonality of interests [Ach98]. In such environments, servers are endowed with much more computing power and have access to much larger bandwidth than clients. Therefore, it becomes cost effective to *push* data from server side rather than follow traditional client-server based

[1] A preliminary version of this appeared in the STACS 2004 conference proceedings

pull model. This can be achieved using *multicast* where a server needs to send a data unit only once to reach arbitrary number of clients [Ste98].

A common way to use multicast in data dissemination is to use server initiated *repetitive* multicast where a server cyclically multicasts data to a large client population. This finds application in many diverse domains, *e.g.*, high-throughput database systems [HGLW87], data management in broadcast disks [Ach98], in solving scalability problems of heavily loaded Web servers [AAF98], content delivery networks (CDNs) [Rab01], etc.

A fundamental question is the order in which the server should multicast data, that is, *scheduling*. In general, clients are seldom interested in individual data items, and attempt to download multiple items. For example, Web clients hardly ever access only one HTML resource, but almost always access the HTML document along with all its embedded images [KR01]. Database clients often access multiple items to complete a read transaction [SNSR99]. Thus client access patterns often show dependencies between consecutive requests, so that the request for a data unit will make it more likely or less likely that a certain data unit will be requested next. These access patterns must be taken into account while designing a good cyclic multicast schedule that has low client-perceived latency while accessing multi-item objects [Lib02b].

One way to model this scenario is to treat the server data set as a weighted directed graph where nodes represent server data units and arc weights represent the strength of the dependency. Then the scheduling problem becomes following question in combinatorial optimization:

Directed Circular Arrangement (DCA) [2]: Given a directed weighted graph $G = (V, E, w)$ with non-negative weights, find a surjection $f : V \mapsto \{0, 1, \ldots, |V| - 1\}$ which minimizes $\sum_{e \in E} w(e)\ell(e)$, where $\ell(e) = (f(v) - f(u)) \bmod n$, for $e = (u, v)$, is called the latency of the edge e in the arrangement f.

1.1 Related Problems

The DCA problem first appeared in the work of Liberatore [Lib02b]. It falls under the class of vertex labelling problems where the question is to find a labelling of the vertices which optimizes some cost function. This class includes many interesting practical problems [Chu81], *e.g.*, optimal linear arrangement problem, directed optimal linear arrangement problem, minimum bandwidth

[2] In keeping with the notation in [SN04] we call this DCA. In the conference version we called it MCA.

problem, folding labelling (also called minimum cut linear arrangement) problem, etc. We give more consideration to optimal linear arrangement problem and directed optimal linear arrangement problem since DCA is closely related to them.

Optimal Linear Arrangement (OLA): Given an undirected weighted graph $G = (V, E, w)$ with non-negative weights, find a surjection $f : V \mapsto \{0, 1, \ldots, n-1\}$ which minimizes $\sum_{e \in E} w(e)\ell(e)$, where $\ell(e) = |f(v) - f(u)|$, for $e = (u, v)$.

OLA problem naturally arose from applications in VLSI design. Garey, Johnson and Stockmeyer [GJS76] proved NP-completeness of the decision version of OLA. Today we know how to solve OLA problem exactly for some special cases of graphs, *e.g.*, un-weighted trees [Shi79,Chu84], outer planar graphs [FH88], cycles, wheels, complete bipartite graphs [JM92], etc. For arbitrary graphs, currently best known guarantee of $O(\log n)$-approximation is due to Rao and Richa [RR98]. Meanwhile, there has also been some work done on polynomial time approximation schemes for un-weighted OLA of dense graphs, namely, [AFK96] and [FK99]. No hardness of approximation results are known for OLA, though it is known that its complimentary problem (Maximizing the cost of a linear arrangement) has a $O(1)$ approximation algorithm.

Directed Optimal Linear Arrangement (DOLA): Given a directed acyclic weighted graph $G = (V, E, w)$ with non-negative weights, find a surjection $f : V \mapsto \{0, 1, \ldots, n-1\}$ such that $(u, v) \in E \implies f(u) < f(v)$, i.e. a topological sort, which minimizes $\sum_{e \in E} w(e)\ell(e)$, where $\ell(e) = f(v) - f(u)$, for $e = (u, v)$.

Not much is known about DOLA. Its decision version was shown to be NP-complete by Even and Shiloach [ES75]. On the algorithmic front, Adolphson and Hu [AH73] gave an $O(n \log n)$-time algorithm to solve DOLA exactly on rooted trees, where all the edges are oriented towards (or away from) the root. The current best approximation algorithm is the $\widetilde{O}(\log n)$-approximation algorithm due to [RR98]. No hardness of approximation results are known for DOLA.

1.2 Current Status

The DCA problem is quite recent [Lib02b]. Only a couple of theoretical results are known about it. Liberatore showed NP-completeness of the weighted version for sparse instances and gave a $\widetilde{O}(\sqrt{n})$ in [Lib02b,Lib02a]. Recently, Naor and Schwartz [SN04] improved the approximation ratio to give an $\widetilde{O}(\log n)$-

approximation algorithm.

1.3 Our Results

In this paper, we start out by proving some preliminary lemmas in section 3 that bound DCA cost. In section 4, we draw comparison between DCA cost and OLA cost (DOLA cost), throwing light on the relative hardness of these problems.

We prove that the decision version of even un-weighted DCA is NP-complete in case of sparse as well as dense graphs (section 5), a stronger result than [Lib02a]. We also consider complementary version of DCA, called MaxCA in section 6. We prove that it is APX-hard and, therefore, has no PTAS unless P=NP. Similarly we show that DCA is APX-hard as well. In section 7, we prove a conditional lower bound of $\sqrt{2} - \epsilon$ for DCA approximation under the assumption that DOLA does not admit constant factor approximation. Finally we conclude with a PTAS for DCA on dense instances in section 9.

2 Notation

By a graph G, we mean a directed graph without parallel edges and loops. V and E, as always, stand for vertex-set and edge-set of G respectively. $|V| = n$. An un-weighted graph is considered as a graph with edges of unit weight. When we talk about the OLA problem on a directed graph we mean the OLA problem on the underlying undirected graph.

Definition 1 *A graph G is **dense** if $|E| = \Omega(n^2)$. More specifically, G is δ-**dense** if $|E| \geq \delta n^2$. Similarly G is **sparse** if $|E| = O(n)$.*

Consider a graph G and $f : V \mapsto \{1, \ldots, n\}$ an arrangement of G.

Definition 2 *An edge $e = (u, v)$ of G is said to be a **forward edge with respect to** f if $f(u) < f(v)$. Similarly e is a **backward edge with respect to** f if $f(u) > f(v)$. Note that the forward/backward status of an edge can be changed by rotating f.*

Let CCOST(f), LCOST(f) and DCOST(f) respectively denote the circular, linear and the directed linear cost of the arrangement f as defined in the problem definitions. For an edge e, $\text{CCOST}_e(f)$ denotes the cost of the edge e in the arrangement f. Similarly for $\text{DCOST}_e(f)$ and $\text{LCOST}_e(f)$. Set $\text{DCOST}_e(f) = \text{DCOST}(f) = \infty$, if any edge $e = (u, v)$ is a backward edge.

Definition 3 *Let g be a circular arrangement of G. By* $\mathrm{ROT}(g)$ *we mean an arrangement h obtained by rotating g so that the total weight of the backward edges is minimized. Note that* $\mathrm{CCOST}(\mathrm{ROT}(g)) = \mathrm{CCOST}(g)$.

Let $\mathrm{DCA}(G)$ be the set of all optimal circular arrangements of G. Similarly define $\mathrm{OLA}(G)$ and $\mathrm{DOLA}(G)$. Note that if G contains a directed cycle then $\mathrm{DOLA}(G) = \emptyset$. Sometimes, by abuse of notation, $\mathrm{DCA}(G)$ also stands for some optimal circular arrangement. Similarly for $\mathrm{OLA}(G)$ and $\mathrm{DOLA}(G)$.

Finally, let $\mathrm{CCOST}(G) := \mathrm{CCOST}(\mathrm{DCA}(G))$. Similarly define $\mathrm{LCOST}(G)$ and $\mathrm{DCOST}(G)$. Again $\mathrm{DCOST}(G) = \infty$ if G contains a directed cycle.

By P_m we mean a directed path $p_1, \ldots, p_{m+1}$ of length m (on $m+1$ vertices), with unit weight edges. By $\overrightarrow{K}_n$ we mean the complete directed acyclic graph on n vertices, i.e. for $1 \le i < j \le n$, there is an edge (i, j) of unit weight.

By $\overleftarrow{G}$ we mean the graph *anti-parallel* to G, that is, $V(\overleftarrow{G}) = V(G)$ and $E(\overleftarrow{G}) = \{(v, u) | (u, v) \in E\}$. The edges in $\overleftarrow{G}$ carry same weight as their counterparts in G.

For graphs G and H, $G + H$ denotes the disjoint union of the two graphs.

3 Bounding DCA Cost

In this section we show some upper and lower bounds on DCA cost and highlight its peculiar features that would help us derive our hardness results.

Proposition 4 *The total weight of the backward edges of* $\mathrm{ROT}(g) \le \mathrm{CCOST}(g)/n$.

PROOF. $\mathrm{CCOST}(g)$ equals the sum taken over all n rotations of g of the total weight of the backward edges. Hence the claim follows by an averaging argument. □

Proposition 5 *Let $G = H_1 + \cdots + H_k$ be a graph with k components. Put $n = |V(G)|$ and $n_i = |V(H_i)|$. Then* $\sum_{i=1}^{k} \mathrm{CCOST}(H_i) \le \mathrm{CCOST}(G) \le n \sum_{i=1}^{k} \mathrm{CCOST}(H_i)/n_i$.

PROOF. For $1 \le i \le k$, let $f_i \in \mathrm{DCA}(H_i)$ be an optimal circular arrangement of H_i. Consider $g = \mathrm{ROT}(f_1) \circ \cdots \circ \mathrm{ROT}(f_k)$. Then each edge of H_i incurs the same latency in g as in $\mathrm{ROT}(f_i)$ except for the backward edges. Also the total weight of the backward edges $\leq \text{CCOST}(H_i)/n_i$ by Proposition 4. Since each such backward edge picks up an additional latency of $n - n_i$ (for all the remaining components), the cost of the H_i edges in the arrangement g is $\leq \text{CCOST}(H_i) + (n - n_i)\frac{\text{CCOST}(H_i)}{n_i}$. This proves the upper bound.

To see the lower bound: Let $g \in \text{DCA}(G)$. If we restrict g to H_i (this may reduce latencies of edges), we get a circular arrangement of H_i. Hence the cost of the H_i edges in $g \geq \text{CCOST}(H_i)$. $\square$

Claim 6 *Proposition 5 is tight.*

PROOF. Let $G = P_{m_1} + \cdots + P_{m_k}$. Then clearly $\text{CCOST}(P_{m_i}) = m_i$ as well as $\text{CCOST}(G) = \sum_i m_i$. This settles tightness of lower bound.

For upper bound, let $G = C_{m_1} + \cdots + C_{m_k}$, where C_m is a directed cycle of length m. Clearly $\text{CCOST}(C_{m_i}) = m_i$. Consider the arrangement obtained by concatenating the optimal ordering for each C_{m_i}. In each cycle, every edge has latency 1, except for one edge which has latency $n - m_i + 1$. Hence the cost of the edges from the cycle C_{m_i} is $m_i - 1 + n - m_i + 1 = n$, giving a total cost of nk, which agrees with upper bound. It remains to show that this is the optimal arrangement.

Let g be any circular ordering of G. Consider any cycle C_{m_i}. As we traverse the edges of the cycle C_{m_i} we will traverse over all the vertices at least once. So the cost of the cycle $C_{m_i} \geq n$. Hence the total cost $\geq nk$. $\square$

This behavior of DCA (with components) enables us to arrive at our hardness results. The fundamental difference between circular (*e.g.* DCA) and linear arrangement problems (*e.g.* OLA, DOLA) is the issue of connectedness. In case of a graph with more than one component it is trivial to the see that the optimal arrangement is obtained by concatenating the optimal arrangements of the individual components. However, such is not the case with circular arrangements. If there are any backward edges in the optimal circular arrangement of one of the components, then the latency of that edge is increased due to the presence of the other components.

Proposition 39 shows a way around this behavior and allows us to assume that the input is a weakly connected graph. We start with some lower bounds on $\text{CCOST}(G)$.

Definition 7 *Let G be a weighted directed graph. For a vertex u, let $w_1 \geq \cdots \geq w_d$ denote the weights of the outgoing edges from u. Define $X^+(u) = \sum_i i w_i$, and $X^+(G) = \sum_{v \in V(G)} X^+(v)$. Similarly define $X^-(u)$ and $X^-(G)$ by replacing outgoing with incoming.*

Proposition 8 CCOST$(G) \geq \max\{X^+(G), X^-(G)\}$.

PROOF. For each vertex u, the best arrangement occurs when the heaviest outgoing edge has latency 1, second heaviest has latency 2 and so on. $X^+(u)$ is precisely the cost of this arrangement. Hence in any arrangement the cost of edges going out of u is $\geq X^+(u)$. Hence we have that CCOST$(G) \geq X^+(G)$. Similarly we also have CCOST$(G) \geq X^-(G)$. □

An easier to handle, but worse lower bound is

Proposition 9 *Let G be an un-weighted directed graph. Then* CCOST$(G) \geq |E|(|E| + n)/2n$, *where $|E|$ denotes the number of edges of G.*

PROOF. Let f be any circular arrangement. For each $1 \leq i \leq n - 1$ there are at most n edges whose latency is i. Thus the best conceivable arrangement is when there are n edges of latency i for $1 \leq i \leq |E|/n$. Choose $k, \delta < n$ such that $|E| = kn + \delta$.

Then

$$
\begin{aligned}
\text{CCOST}(f) &\geq n \sum_{i=1}^{k} i + (k+1)\delta \\
&= \frac{[(k+1)n][kn + 2\delta]}{2n} \\
&= \frac{(|E| + n - \delta)(|E| + \delta)}{2n} \\
&\geq \frac{|E|(|E| + n)}{2n}
\end{aligned}
$$

Hence the result. □

For weighted graphs, the same proof yields,

Proposition 10 *Let G be a weighted directed graph. Let $w_1 \geq \cdots \geq w_{|E|}$ be the weights of the edges. Then* CCOST$(G) \geq \sum_i w_i \lceil i/n \rceil$.

It is easy to see that the lower bound given by Proposition 10 is worse than Proposition 8.

We now give a lower bound for CCOST(G) in terms of the optimal cost on a subgraph.

Proposition 11 *Let G be a weighted directed graph and $E = E_0 \cup E_1$ be a partition of it. Let $G_i = (V(G), E_i)$ be the subgraphs obtained by removing the edges E_{1-i} from G. Then* $\mathrm{CCOST}(G) \geq \mathrm{CCOST}(G_0) + \mathrm{CCOST}(G_1)$.

PROOF. Let g be any optimal arrangement of G. Removing the edges in E_i gives an arrangement g_i of G_i. Thus $\mathrm{CCOST}(g_i) \geq \mathrm{CCOST}(G_i)$. Since the edges in G_0 and G_1 are disjoint, we have $\mathrm{CCOST}(G) = \mathrm{CCOST}(g) = \mathrm{CCOST}(g_0) + \mathrm{CCOST}(g_1) \geq \mathrm{CCOST}(G_0) + \mathrm{CCOST}(G_1)$. □

Corollary 12 *Let G be a weighted directed graph and $U \subseteq V(G)$ an independent set. Then* $\mathrm{CCOST}(G) \geq \mathrm{CCOST}(G') + \sum_{u \in U}(X^+(u) + X^-(u))$. *Here G' is the graph obtained by removing all the edges incident to U but keeping the vertices in U.*

PROOF. Let E_0 be the set of edges incident to U, and E_1 denote the remaining edges. Let G_0 and G_1 be as in Proposition 11. Thus we have $\mathrm{CCOST}(G) \geq \mathrm{CCOST}(G_0) + \mathrm{CCOST}(G_1)$. The graph G_1 is just G'.

The graph G_0 is just a collection of isolated points and stars centered at the vertices in U (since U is an independent set). Thus the optimal arrangement in this case is the concatenation of the optimal arrangements of the stars (there are no backward edges in the optimal arrangements) and the isolated vertices. The cost of the optimal arrangement of the star containing $u \in U$ is precisely $X^-(u) + X^+(u)$.

Hence the result. □

4 Comparison of DCA with OLA and DOLA

Now we are ready to compare DCA with OLA and DOLA.

4.1 Comparison with OLA

Proposition 13 *For any graph G,* $\mathrm{LCOST}(G) \leq 2(1 - 1/n) \cdot \mathrm{CCOST}(G)$.

PROOF. Let $f \in \mathrm{DCA}(G)$ be an optimal circular arrangement. Denote by f_i the arrangement got by rotating f by i-positions. Thus $f_0 = f$. Clearly $\mathrm{CCOST}(f_i) = \mathrm{CCOST}(f)$ for all i. We now show that for some i, $\mathrm{LCOST}(f_i) \leq (2 - 1/n) \cdot \mathrm{CCOST}(f)$.

Consider any edge $e = (u, v)$ of weight w with latency p with respect to the f ordering. The cost of this edge in the linear arrangement f_i is pw if $f_i(u) < f_i(v)$ and $(n-p)w$ if $f_i(u) > f_i(v)$. Hence

$$\begin{aligned}\sum_i \mathrm{LCOST}_e(f_i) &= (n-p)\cdot pw + p\cdot(n-p)w \\ &= 2p(n-p)w \\ &= 2(n-p)\,\mathrm{CCOST}_e(f)\end{aligned}$$

$$\frac{1}{n}\sum_i \mathrm{LCOST}_e(f_i) \le 2(1-1/n)\,\mathrm{CCOST}_e(f)$$

$$\sum_e \frac{1}{n}\sum_i \mathrm{LCOST}_e(f_i) \le 2(1-1/n)\sum_e \mathrm{CCOST}_e(f)$$

$$\frac{1}{n}\sum_i \mathrm{LCOST}(f_i) \le 2(1-1/n)\,\mathrm{CCOST}(f)$$

Hence there is some i for which $\mathrm{LCOST}(f_i) \le 2(1-1/n)\,\mathrm{CCOST}(f)$. □

To see that the above result is tight, consider $G = C_n$ a directed cycle. Clearly the optimal circular arrangement has cost n. Let u and v denote the first and last vertices of an optimal linear arrangement of C_n. Then we have $\mathrm{LCOST}(C_n)$ is the sum of the cost of the path $u \rightsquigarrow v$ and that of $v \rightsquigarrow u$. By triangle inequality we see that the cost of $u \rightsquigarrow v$ is at least $n-1$. The same holds for $v \rightsquigarrow u$. Hence $\mathrm{LCOST}(C_n)$ is at least $2(n-1)$. It is achieved by the obvious arrangement. This gives a ratio of $2(1-1/n)$.

Another observation is that $\mathrm{LCOST}(G)$ does not depend on the orientation of the edges of G, while $\mathrm{CCOST}(G)$ depends on the orientation. So we have actually proved that $\mathrm{LCOST}(G) \le (2-1/n)\,\mathrm{CCOST}(H)$, where H is G with any orientation.

To bound the circular cost by in terms of the optimal linear cost, we note that in any arrangement (circular or linear) each edge has latency at least 1 and at most $n-1$. Thus we have $\mathrm{CCOST}(G) \le (n-1)\,\mathrm{LCOST}(G)$. Even though this is a trivial inequality, there are examples which almost achieve this bound. In fact,

Claim 14 *There is a family of graphs G_n which has n vertices for which $\mathrm{CCOST}(G) \ge (n/6)\,\mathrm{LCOST}(G)$.*

PROOF. Let G have vertices $0, 1, \ldots, n-1$. The edges of G are $\{(i, i+1) : 0 \le i \le n-1\} \cup \{(i+1, i-1) : 0 \le i \le n-1\}$, where addition is done modulo

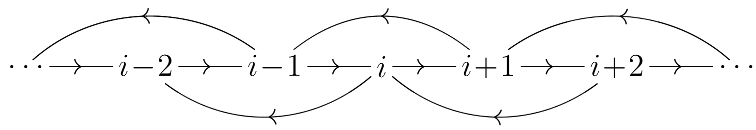
n. G is essentially a sunflower with the base circle and the petals oriented in opposite directions as shown in Figure 1. Considering the obvious arrangement shows that $\text{LCOST}(G) \leq 3n$ (n edges of latency 1, n edges of latency 2).

On the other hand, consider any optimal circular arrangement f: For each i, consider the vertices $i-1, i, i+1$, and the three edges $(i-1, i), (i, i+1), (i+1, i-1)$. The total cost of these three edges is n if they appear in clockwise order and $2n$ otherwise.

Summing this over all i, we have $\text{CCOST}(G) \geq n^2/2$ (since each edge is counted at most twice). Hence the result. □

In fact, it can be easily shown that the optimal cost is $n(n+2)/2$ if n is even and $n(n+1)/2$ otherwise.

Fig. 1. Sunflower on n vertices...



Note that for this graph $X^+(G) = X^-(G) = O(n)$ while $\text{CCOST}(G) \geq n^2/2$. This shows that the lower bound of Proposition 8 is far from tight.

4.2 Comparison with DOLA

From the definition, any legal DOLA arrangement is a legal DCA arrangement. Hence we trivially have $\text{CCOST}(G) \leq \text{DCOST}(G)$. On the other hand, $\text{DCOST}(G)$ is trivially $\leq (n-1)\,\text{CCOST}(G)$. In case of weighted graphs this is optimal as shown in [Lib02b]. For completeness we reproduce that example here.

Let G_n be a directed path of length $n-1$ together with the edge (p_1, p_n) of weight W. The only feasible DOLA arrangement f is $\{1, \ldots, n\}$. Hence $\text{DCOST}(G_n) = (W+1)(n-1)$. Considering the arrangement $\{1, n, 2, \ldots, n-1\}$ shows that $\text{CCOST}(G_n) \leq W + (n-3) + 2 + 2 = W + n + 1$. If we set $W = n^2$, we see that $\text{DCOST}(G_n)/\,\text{CCOST}(G_n) \sim n-1$.

Even if we restrict ourselves to un-weighted graphs, we can still get a ratio of $\Omega(n^{2/3})$.

Claim 15 *There is a family of un-weighted directed graphs G_n for which* $\text{DCOST}(G_n)/\,\text{CCOST}(G_n) > n^{2/3}/3$.

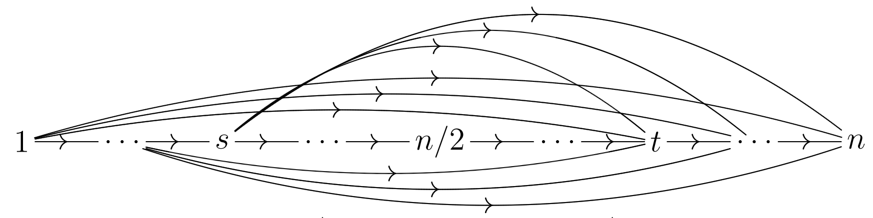


Fig. 2. Example showing $\mathrm{DCOST}(G)/\,\mathrm{CCOST}(G) = \Omega(n^{2/3})$

PROOF. Put $s = n^{1/3}, t = n+1-s$. Start with the graph P_{n-1}. For $1 \le i \le s$ and for $t \le j \le n$ put an edge (i, j). This defines G_n (see Figure 2). There is only one feasible DOLA arrangement, and it has cost

$$n - 1 + \sum_{i=1}^{s}\sum_{j=t}^{n}(j-i) \sim n - 1 + \sum_{i=1}^{s}[s(n+t)/2 - si] \sim s^2 n = n^{5/3}.$$

On the other hand, considering the arrangement $1, \ldots, s, t, \ldots, n, s+1, \ldots, t-1$ gives an DCA cost of

$$\sim n + 2s + \sum_{i=1}^{s}\sum_{j=1}^{n+1-t}(s+j-i) \sim n + 2s + s^3 < 3n.$$

This gives the ratio as claimed. $\square$

We now show an upper bound on $\mathrm{DCOST}(G)$.

Proposition 16 *Let G be an un-weighted directed graph and f any DOLA arrangement of G. Then*

$$\mathrm{DCOST}(f) \le |E|n - \frac{2\sqrt{2}}{3}|E|\sqrt{|E|} + \frac{7}{3}|E|$$

Moreover for interesting E (i.e. $|E| \ge 28$), $\mathrm{DCOST}(f) \le |E|n - |E|\sqrt{|E|}/2$.

PROOF. Assume without loss of generality that the underlying undirected graph is connected. For each $1 \le i \le n-1$, there can be at most $n-i$ edges with latency i. Thus the worst case cost is achieved when we have 1 edge of latency $n-1$, 2 edges of latency $n-2$ ... till $k-1$ edges of latency $n-k+1$ and δ edges of latency $n-k$, where $\binom{k}{2}+\delta = |E|$. Thus $\mathrm{DCOST}(f)$ is bounded above by $\sum_{i=1}^{k-1} i(n-i) + \delta(n-k) = n\binom{k}{2} - (k-1)(k)(2k-1)/6 + \delta(n-k)$. Note that $\delta \leq k-1 \implies |E| < \binom{k+1}{2}$.

$$\begin{aligned}
\mathrm{DCOST}(f) &= n\binom{k}{2} - \binom{k+1}{2}\frac{(k-1)(2k-1)}{3(k+1)} + \delta(n-k) \\
&\leq n\left[\binom{k}{2} + \delta\right] - \binom{k+1}{2}\frac{2k-5}{3} - k\delta \\
&\leq n|E| - |E|\frac{2(k+1)-7}{3} \\
&\leq n|E| - |E|\sqrt{|E|}\frac{2\sqrt{2}}{3} + \frac{7|E|}{3}
\end{aligned}$$

The first inequality follows from the fact that $(k-1)(2k-1) = 2k^2 - 3k + 1 > 2k^2 - 3k - 5 = (k+1)(2k-5)$. This completes the first claim. If $|E| \geq 28$, then

$$\left(\frac{2\sqrt{2}}{3} - \frac{1}{2}\right)\sqrt{|E|} \geq \frac{7}{3}$$

thus proving the second claim. □

On the same lines one can show a lower bound on DCOST(G). In this case, we choose $n-1$ edges with latency 1, $n-2$ edges with latency 2,.... Hence $\mathrm{DCOST}(G) \geq \sum_{i=1}^{k-1} i(n-i)$, where k is chosen so that $nk - \binom{k+1}{2} \leq |E| \leq n(k+1) - \binom{k+2}{2}$. The lower bound of Proposition 8 also holds for DOLA arrangements.

Both these bounds can be extended to the weighted case as well. We conclude this section with an upper bound on the DCOST / CCOST ratio.

Corollary 17 *For any graph G,*

$$\mathrm{DCOST}(G) \leq \mathrm{CCOST}(G)\frac{n(2n - \sqrt{|E|})}{|E| + n}.$$

In particular if

- $|E| = \Omega(n^{1+\epsilon})$ *for* $\epsilon < 1$, *then* $\mathrm{DCOST}(G)/\mathrm{CCOST}(G) = O(n^{1-\epsilon})$.
- G *is* δ^2*-dense then* $1 \leq \mathrm{DCOST}(G)/\mathrm{CCOST}(G) \leq (2-\delta)/\delta^2$.

PROOF. From Proposition 16 and Proposition 9, we have

$$\frac{\mathrm{DCOST}(G)}{\mathrm{CCOST}(G)} \leq \frac{|E|n - |E|^{3/2}/2}{|E|(|E|+n)/(2n)}$$
$$= n\frac{2n - \sqrt{|E|}}{|E|+n}.$$

The remaining claims follow by substituting appropriate values in the general upper bound. □

The above proof shows that in order to get a $\Omega(n)$ separation between DCOST(G) and CCOST(G), we need to look only at sparse graphs. Moreover any approximation algorithm for DOLA on dense graphs automatically yields an approximation algorithm for DCA on dense graphs. However an approximation algorithm for DCA does not apriori give rise to a DOLA approximation algorithm since an DCA arrangement need not be a legal DOLA arrangement.

Does DOLA admit PTAS for dense instances?

4.3 Undirected DCA variant

Another variant of DCA considered in [Lib02a] is:

Undirected DCA: Given an un-directed weighted graph $G = (V, E, w)$ with non-negative weights, find a surjection $f : V \mapsto \{0, 1, \ldots, n-1\}$ which minimizes $\sum_{e \in E} w(e)\ell(e)$, where $\ell(e) = \min\,[(f(v) - f(u) \mod n, f(u) - f(v) \mod n]$, for $e = (u, v)$. Let UCCOST(f) denote the cost of an arrangement f.

Liberatore [Lib02a] gives an $O(\log n)$-approximation algorithm for the same. His algorithm is based on a divide-and-conquer strategy. We dispose of the undirected DCA variant by exhibiting two simple functions $X(G, f)$ and $Y(G, g)$ such that, for any undirected weighted graph G,

- if f is an OLA arrangement, then $X(G, f)$ is an undirected DCA arrangement with $\mathrm{UCCOST}(X(G, f)) \leq \mathrm{LCOST}(f)$, and
- if g is an undirected DCA arrangement, then $Y(G, g)$ is an OLA arrangement with $\mathrm{LCOST}(Y(G, g)) \leq 2 \cdot \mathrm{UCCOST}(g)$.

Now it clearly follows that any approximation algorithm for OLA (in particular $O(\log n)$-approximation algorithm of Rao and Richa [RR98]) is also an approximation algorithm for undirected DCA with twice the approximation ratio.

Set $X(G, f) = f$ and $Y(G, g) = g'$, where g' is the cheapest *linear* arrangement among the n different rotations of g. The fact that $\text{UCCOST}(f) \leq \text{LCOST}(f)$ follows from definition and $\text{LCOST}(g') \leq 2\,\text{UCCOST}(g)$ is proved similar to Proposition 13.

5 NP Completeness

Theorem 18 (Proposition 3.1 in [Lib02a]) *The decision version of the DCA problem is NP-complete.*

Liberatore [Lib02a] proves that *weighted* DCA problem is NP-complete by a reduction from an un-weighted DOLA. Since the DCA instance in his proof has $|E| = O(|V|)$, we infer that DCA is NP-complete even when restricted to sparse graphs. In this section we prove that even *un-weighted* DCA is NP-complete in case of sparse as well as dense graphs, stronger result than Liberatore [Lib02a]. We too make use of reduction from an un-weighted DOLA.

5.1 Straightening Algorithm

We start with an algorithm which allows us to normalize optimal solutions in a special case.

Theorem 19 (Straightening Algorithm) *Let G be a weighted directed graph, and $m > 2$. Let f be any circular arrangement of $G + P_m$. We can transform f (in time polynomial in $m + n$) to an arrangement g in which all the vertices in P_m appearing in the order $p_1, \ldots, p_{m+1}$. Moreover* $\text{CCOST}(g) \leq \text{CCOST}(f)$.

PROOF.

Let f be any circular arrangement of $G + P_m$. We define a sequence of arrangements $g_1, \ldots, g_{m+1}$ with the following properties:

- $g_i(j) = p_j$ for all $1 \leq i \leq m+1, 1 \leq j \leq i$
- $\text{CCOST}(g_{i+1}) \leq \text{CCOST}(g_i)$ for $1 \leq i \leq m$

Thus $g = g_{m+1}$ is the required arrangement. To start, let $g_1 = f$ suitably rotated so that $g_1(1) = p_1$. Note that $\text{CCOST}(g_1) = \text{CCOST}(f)$. Assume we know g_i and $i \leq m$ (else we are done). If $g_i(i+1) = p_{i+1}$, then set $g_{i+1} = g_i$ and continue with the next i.

Suppose $g_i(i+1) \neq p_{i+1}$. Let $i + \ell$ denote the position of the vertex p_{i+1} ($2 \leq \ell \leq m + n - i$). Partition the vertices as follows: $L = \{p_1, \ldots, p_i\}$, $M = \{g_i(i+1), \ldots, g_i(i+\ell-1)\}$, $R = \{g_i(i+\ell+1), \ldots, g(m+n+1)\}$. Thus the arrangement g_i is $L\, M\, p_{i+1}\, R$.

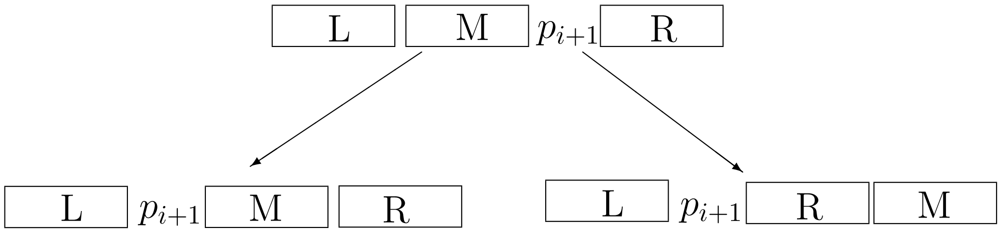


Fig. 3. Possible rearrangements: A (Left) and B (Right) from g_i (Top)

Let A be the arrangement $L\, p_{i+1}\, M\, R$ and B be the arrangement $L\, p_{i+1}\, R\, M$ as shown in Figure 3. Basically A is the order obtained by moving p_{i+1} to position $i+1$ and pushing everybody else over by one. B is obtained from A by swapping the M and R parts of the arrangement.

The tables in Figure 4 below show the different types of possible edges and how the cost of those edges change when moving from g_i to A and B. 0 indicates no change in cost, $-$ indicates no such edge exists, $\downarrow$ indicates a possible decrease (decrease or same) in cost and $\uparrow$ indicates a possible increase in cost.

	L	M	p_{i+1}	R
L	0	$-$	$\downarrow$	$-$
M	$-$	0	$-$	$\downarrow$
p_{i+1}	$-$	$\downarrow$	$-$	$\uparrow$
R	$-$	$\uparrow$	$-$	0

	L	M	p_{i+1}	R
L	0	$-$	$\downarrow$	$-$
M	$-$	0	$-$	$\uparrow$
p_{i+1}	$-$	$\downarrow$	$-$	$\uparrow$
R	$-$	$\downarrow$	$-$	0

Fig. 4. Moving from g_i to A (left) and B (right) [edges from ROW to COLUMN]

Some observations: Since L is a prefix of the path, it has no incoming edges, and the only outgoing edge is to p_{i+1}. The successor of p_{i+1} (i.e. p_{i+2}) may be in the M or the R segment.

Let W_{MR} be the total weight of all the edges going from M to R and W_{RM} be the total weight of all edges going from R to M.

Case 1: $W_{MR} \geq W_{RM}$: We claim that $\text{CCOST}(A) \leq \text{CCOST}(g_i)$. From Figure 4, there are two costs associated with the edges from p_{i+1}, viz. $e_1 := (p_i, p_{i+1})$ and $e_2 := (p_{i+1}, p_{i+2})$. The latency of e_1 decreases by $\ell - 1$ (from ℓ to 1). If $p_{i+2} \in R$, then the latency of e_2 increases by $\ell - 1$ (t to $\ell + t - 1$, where

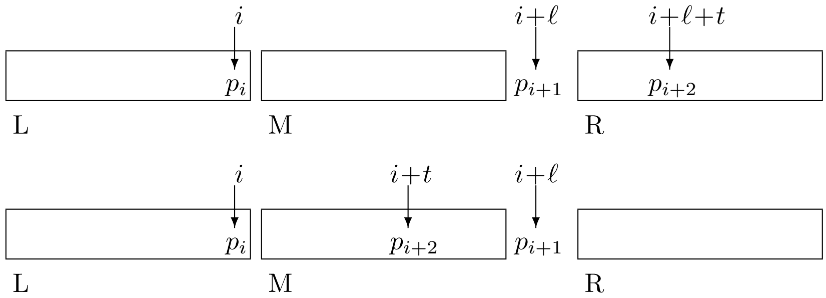


Fig. 5. Possibilities for p_{i+2} in g_i: To the right (top) or to the left of p_{i+1} (bottom)

$i + \ell + t$ is position of p_{i+2}). On the other hand if $p_{i+2} \in M$ at position $i + t$ $(1 < t < \ell)$, then latency changes from $m+n-((i+\ell)-(i+t)) = m+n+t-\ell$ to $((i+t+1)-(i+1)) = t$. Since $m+n+t-\ell \geq t$ $(\ell \leq m+n)$, the latency only decreases, leading to a decrease in cost.

It remains to consider the edges between M and R. All the edges from M to R, loose latency of 1 each, leading to decrease of W_{MR} in the total cost. Similarly all edges from R to M gain latency on 1 each, leading to an increase of W_{RM}. Since $W_{MR} \geq W_{RM}$, the net cost does not increase. Hence $\mathrm{CCOST}(A) \leq \mathrm{CCOST}(g_i)$.

Case 2: $W_{MR} \leq W_{RM}$: We claim that $\mathrm{CCOST}(B) \leq \mathrm{CCOST}(g_i)$ in this case. Again like before let $e_1 := (p_1, p_{i+1})$ and $e_2 := (p_{i+1}, p_{i+2})$. The latency of the edge e_1 decreases by $\ell - 1$ (from ℓ to 1). If $p_{i+2} \in R$, then latency of e_2 remains the same. If $p_{i+2} \in M$ at position $i+t$, then the latency changes from $m+n-((i+\ell)-(i+t)) = m+n+t-\ell$ to $|R|+t = (m+n)-(i+\ell)+t = m+n+t-\ell-i$. Thus the latency of the edge e_2 decreases in this case.

Now consider any edge (x, y) from M to R. Latency in g_i: $x \mapsto$ (end of M) $\mapsto$ (start of R) $\mapsto y$. In the arrangement B: $x \mapsto$ (end of M) $\mapsto$ (start of L) $\mapsto$ (end of L) $\mapsto p_{i+1} \mapsto$ (start of R) $\mapsto y$. So the latency of each edge increases by $i+1$ (length of L + 1). Hence the cost increases by $(i+1)W_{MR}$. Similarly, an edge (x, y) from R to M saves a latency of exactly $i+1$, leading to a decrease in cost of $(i+1)W_{RM}$. Since $W_{MR} \leq W_{RM}$ the total cost does not increase. Hence $\mathrm{CCOST}(B) \leq \mathrm{CCOST}(g_i)$.

Now set $g_{i+1} = A$ if $W_{MR} \geq W_{RM}$ and $g_{i+1} = B$ otherwise. This completes the construction of the sequence. Clearly this construction leads to a polynomial time algorithm. $\square$

In fact the time complexity of the straightening algorithm is $O(mn^2)$ if implemented in the naivest fashion as given. More efficient implementation would first consider restriction of f to only G, that is $f|G$, find $\mathrm{ROT}(f|G)$, and then output $g = p_1p_2 \ldots p_{m+1} \circ \mathrm{ROT}(f|G)$ requiring $O(m+n^2)$ time. A generalization of this is proved in [Lib02a].

Proposition 20 (Lemma 3.4 of [Lib02a]) *Let* $G = H_1 + \cdots + H_k$ *be a directed graph with* k *components. Then there is an optimal circular arrangement if* G*, which can be obtained by concatenating circular arrangements of* H_i*.*

PROOF. [Sketch] Let $g \in \mathrm{DCA}(G)$. Let f_i be the arrangement of H_i got by restricting g to vertices of H_i (this may reduce latencies of the edges). Then the arrangement $g' = \mathrm{ROT}(f_1) \circ \cdots \circ \mathrm{ROT}(f_k)$ is a circular arrangement of G and more over $\mathrm{CCOST}(g') \leq \mathrm{CCOST}(g)$. But optimality of g implies $\mathrm{CCOST}(g') = \mathrm{CCOST}(g)$. □

We now have a corollary of Theorem 19 which gives us a technique to force an optimal DCA arrangement to have only forward edges.

Corollary 21 *Let* G *be an un-weighted directed acyclic graph, and* $m \geq \mathrm{DCOST}(G)$*. Let* g *be the circular arrangement obtained by concatenating* P_m *with the optimal DOLA arrangement of* G*. Then* $\mathrm{CCOST}(G + P_m) = \mathrm{DCOST}(G) + m = \mathrm{CCOST}(g)$*, i.e.* g *is an optimal circular arrangement.*

PROOF. First note that $\mathrm{CCOST}(g) = \mathrm{DCOST}(G) + m \leq 2m$. Now let f be any optimal circular arrangement of $G + P_m$. Apply the straightening algorithm to f to get a circular arrangement h where all vertices of P_m appear in order. $\mathrm{CCOST}(h) = \mathrm{CCOST}(f)$, since f is optimal and the straightening algorithm cannot increase the cost.

Clearly both h and g start with the vertices of P_m in the same order. We now show that G cannot have any backward edges in the h arrangement. Suppose to the contrary. Let e be an backward edge of G with respect to h. Then the latency of $e > m$ because of the intervening P_m. Thus the cost of the arrangement $h > 2m$ (m for cost of P_m edges and m for e). By assumption on m, we have $2m \geq \mathrm{CCOST}(g)$, contradicting optimality of h. Thus G has no backward edges in the arrangement h. Hence the arrangement h restricted to G is a DOLA arrangement, and must be optimal. In particular $\mathrm{CCOST}(h) = \mathrm{CCOST}(G + P_m) = \mathrm{DCOST}(G) + m = \mathrm{CCOST}(g)$. □

We conclude this section with a couple of NP Completeness proofs of un-weighted DCA. All these results extend to the case when all edge weights are bounded above by a fixed polynomial.

Theorem 22 *The decision version of the un-weighted DCA problem is NP-complete.*

PROOF. Proof by reduction from un-weighted DOLA. Let (G, K) be a DOLA instance. Let $m = n^3$ be an upper bound for cost of optimal DOLA arrangement. By Corollary 21, $\text{CCOST}(G + P_m) = DOLA(G) + m$. So if $G' = G + P_m$ and $K' = K + m$, we have $DOLA(G) \leq K \iff \text{CCOST}(G + P_m) \leq K'$. $\square$

Since the DCA instance in this proof has $|E| = O(|V|)$, we infer that un-weighted DCA is NP-complete even when restricted to sparse graphs. We now prove a generalization of Corollary 21 and use it to show that un-weighted DCA is NP-complete even when restricted to dense instances.

Proposition 23 *Let G and H be un-weighted directed acyclic graphs such that $|V(H)| = m \geq \text{DCOST}(G)$. Assume further that there is an optimal DCA arrangement h of H which does not contain any backward edges. Let g be the circular arrangement obtained by concatenating h with the optimal DOLA arrangement of G. Then $\text{CCOST}(G + H) = \text{DCOST}(G) + \text{DCOST}(H) = \text{CCOST}(g)$, i.e. g is an optimal circular arrangement.*

PROOF. Let f be any optimal circular arrangement of $G+H$. Apply Proposition 20 to f to get a circular arrangement f' where all vertices of H appear in order. Since h is an optimal DCA arrangement of H without any backward edges, we can replace the H part of the arrangement f' with h without changing the optimality of f'.

Now proceed as in Corollary 21 to show that G cannot have any backward edges in the arrangement f'. Hence $\text{CCOST}(G + H) = \text{CCOST}(f) = \text{CCOST}(f') = \text{DCOST}(G) + \text{DCOST}(H) = \text{CCOST}(g)$. $\square$

Theorem 24 *Decision version of the un-weighted DCA is NP-complete even when restricted to dense instances.*

PROOF. Proof by reduction from un-weighted DOLA. Let (G, K) be a DOLA instance. Let $m = n^3$ be an upper bound for cost of optimal DOLA arrangement. Now consider DCA instance $G' = G + \overrightarrow{K}_m$, where $\overrightarrow{K}_m$ is the complete DAG on m vertices. Note that the total number of vertices in G' is $m + n = O(m)$ and the number of edges is $\Omega(m^2)$, that is, it is *dense.*

Let f denote the obvious arrangement of $\overrightarrow{K}_m$, i.e. $1, \ldots, m$. Since $\text{CCOST}(f) = \binom{m+1}{3} = X^+(\overrightarrow{K}_m)$ (refer to Proposition 8), f is optimal. Moreover f does not have any backward edges. Hence $\overrightarrow{K}_m$ satisfies the hypotheses of Proposition 23.

Therefore, $\text{CCOST}(G') = DOLA(G) + \binom{m+1}{3}$. Letting $K' = K + \binom{m+1}{3}$, we have $DOLA(G) \leq K \iff \text{CCOST}(G') \leq K'$. $\square$

We now show that unweighted DCA is NP-complete when restricted to arbitrary given instances. More specifically,

Theorem 25 *Let $M : \mathbb{N} \mapsto \mathbb{N}$ be any polynomial time computable function such that for large t, $t \leq M(t) \leq t^2/2$. Call an instance (V, E) of DCA M-dense if*

$$||E| - M(|V|)| \leq |V|$$

Decision version of the un-weighted DCA is NP-complete even when restricted to M-dense instances.

PROOF. The proof is similar to that of Theorem 24, a reduction from DOLA. Given a DOLA instance $G = (V, E)$, let $t = |V|^3 + 1$, be an upper bound on the DOLA cost. Depending on the value of $M(t)$, we construct a graph H with the following properties

- H is "between" P_t and $\overrightarrow{K}_t$.
- The number of edges in H is about $M(t)$.
- H admits a unique topological sort h.
- $\text{DOLA}(H) = \text{DCA}(H) = h$.

Number the vertices of H, $1, \ldots, t$. For a parameter k, put a directed edge from vertex i to j if $i < j \leq i + k$. In particular it contains a directed path of length $t-1$. Choose k so that the total number of edges is between $M(t) - t/2$ and $M(t) + t/2$. The only topological ordering of H is $h = 1, \ldots, t$. On the other hand, $\text{CCOST}(h)$ achieves the lower bound of Proposition 8. Thus h is both the optimal DOLA as well as DCA ordering.

Now we proceed like in Theorem 24. From Proposition 23, we have $\text{CCOST}(G + H) = \text{DCOST}(G) + \text{DCOST}(H)$. Since we know the exact value of $\text{DCOST}(H)$, we have that $\text{CCOST}(G + H) \geq \alpha \iff \text{DCOST}(G) \geq \alpha - \text{DCOST}(H)$. Hence the result.

6 Hardness of Approximation

Papadimitriou and Yannakakis [PY91] defined a syntactic class of NP optimization problems, and showed that the complete problems (under L-reductions) for this class are all equally hard to approximate. The celebrated PCP theorem [ALM+98] was then used to show that if any MAX-SNP complete problem admitted a PTAS then P=NP.

APX-PB is the class of NP optimization problems which admit constant factor approximation algorithm and have polynomial bounded cost. It was shown in [KMSV98] that the closure of MAX-SNP under E-reductions equals APX-PB. E-reductions are generalizations of L-reductions.

In this section we show that the DCA problem does not admit any PTAS unless P=NP.

MAX SUBDAG: Given a directed graph $G = (V, E)$, find a subset $E' \subseteq E$ of maximum cardinality for which (V, E') is acyclic.

Let $F^* \subseteq E$ denote the optimal solution to the MAX SUBDAG problem. Then,

Proposition 26 *$|F^*| \geq |E|/2$.*

PROOF. Consider any arrangement f of V. Either f or its reverse ordering must have more than $|E|/2$ forward edges. □

Definition 27 *Let I be an instance of an optimization problem Π, and S any feasible solution. The **relative error of solution S for instance I***

$$\mathcal{R}(S, I) := \frac{|Cost(OPT(I)) - Cost(S)|}{Cost(OPT(I))}$$

where $Cost$ is the cost function and $OPT(I)$ is the optimal solution for the instance I.

Before venturing into the proof, let's briefly review notion of E-reduction from (reference). Consider two optimization (maximization or minimization) problems Π and Π'. We say that Π E-reduces to Π' if there are two polynomial-time algorithms $\mathcal{A}$, $\mathcal{B}$ and a polynomially bounded function p, such that for each instance I of Π:

(1) Algorithm $\mathcal{A}$ produces an instance $I' = \mathcal{A}(I)$ of Π', such that the optima of I and I', satisfy $\mathrm{OPT}(I') \leq p(|I|)\,\mathrm{OPT}(I)$.
(2) Given any solution S' of I', algorithm $\mathcal{B}$ produces a solution of S of I such that the relative errors of the two solutions satisfy

$$\mathcal{R}(S, I) \leq \mathcal{R}(S', I')$$

We demonstrate suitable choice of algorithms $\mathcal{A}$ and $\mathcal{B}$ and polynomial p, and prove the following theorem.

Theorem 28 *MAX SUBDAG is E-reducible to DCA.*

Algorithm $\mathcal{A}$: Let I be an instance of MAX SUBDAG problem with un-weighted graph $G = (V, E)$ as input. Then consider un-weighted graph G', where $G' = G + P_m$, $m = n^4$.

Given any solution $S \subseteq E$ for the MAX SUBDAG problem, define an arrangement $X(S)$ as follows: $X(S)$ consists of a topological sort of (V, S) followed by the vertices of P_m in canonical order.

Conversely, given any arrangement f for G' define $Y(f) \subseteq E$ as follows: Apply the straightening algorithm to f to get an arrangement g (of possibly lower cost). $Y(f)$ is the set of all forward edges of E in the arrangement g.

Algorithm $\mathcal{B}$: Given an arrangement f of G', return $Y(f) \subseteq E$.

Let $S \subseteq E$ be any solution for MAX SUBDAG (not necessarily optimal). Let $f = X(S)$. Then every edge in S has cost $\leq n$, and every edge in $E \setminus S$ has cost $\leq (m + n)$. Thus

$$\mathrm{CCOST}(X(S)) \leq |S|n + |E \setminus S|(m + n) + m \leq (|E| + 1)m$$

Proposition 26 implies $|S| \geq |E|/2$. Hence we have $\mathrm{OPT}(I') \leq (2\,\mathrm{OPT}(I) + 1)m \leq 3m\,\mathrm{OPT}(I)$. So we can take $p(n) = 3m$.

Let f be any solution for I' and f' be the arrangement obtained by straightening f, and $S = Y(f)$ be the forward edges of f' in E.

$$|F| + (|E| - |F|)m + m \leq \mathrm{CCOST}(f') \leq \mathrm{CCOST}(f) \leq (1 + \epsilon)\,\mathrm{CCOST}(G').$$

Now define an arrangement f^* from F^* as follows: Take a topological sort of (V, F^*) and add the vertices of P_m. Note that F^* is the set of forward edges of G in the arrangement f^*. Then,

$$\text{CCOST}(G') \leq \text{CCOST}(f^*) \leq n|F^*| + (m+n)(|E| - |F^*|) + m,$$

since each edge in F^* has latency $< n$ and each edge in $E \setminus F^*$ has latency $< m+n$. Combining the two inequalities, we get

$$|F| + m(|E| - |F|) + m \leq (1+\epsilon) \cdot [n|F^*| + (m+n)(|E| - |F^*|) + m].$$

Algebraic simplifications reduce it to

$$(1+\epsilon) \cdot |F^*| - \epsilon - |E| \cdot \left(\epsilon + \frac{(1+\epsilon)n}{m}\right) \leq \frac{m-1}{m} \cdot |F|.$$

Since $|F^*| \geq |E|/2$, $|F^*| \geq 1$, $\epsilon < 1/3$, and $n/m = \epsilon/4$, we get

$$\begin{aligned}
\frac{|F|}{|F^*|} &\geq \frac{m}{m-1}\left[1 + \epsilon - \epsilon - 2\left(\epsilon + \frac{(1+\epsilon)n}{m}\right)\right] \\
&> \left[1 - 2\epsilon - \frac{2(1+\epsilon)n}{m}\right] \\
&> (1 - 3\epsilon).
\end{aligned}$$

Thus we have a $(1-3\epsilon)$-approximation algorithm for MAX SUBDAG. This is a contradiction unless P=NP.

7 Hardness of Approximation

We now turn to hardness of approximation and start by proving a curious hardness result for DCA.

Proposition 29 *Suppose that un-weighted DOLA has a polynomial time α-approximation algorithm and un-weighted DCA has a polynomial time $(1+\delta)$-approximation algorithm ($\delta < 1$). Then un-weighted DOLA has a polynomial time β-approximation algorithm, where*

$$\beta = (1+\delta)\left[1 + \frac{\alpha\delta}{1-\delta}\right].$$

PROOF. Let $\mathcal{D}$ denote the α-approximation algorithm for un-weighted DOLA, and $\mathcal{M}$ denote the $(1+\delta)$-approximation algorithm for un-weighted DCA. Let $\mathcal{SA}$ denote the straightening algorithm of Theorem 19. Put $\theta = (1+\delta)/(1-\delta)$.

Require: Input un-weighted directed graph G.
Ensure: g is a β-approximate DOLA arrangement of G.
1: $\texttt{f} \Leftarrow \mathcal{D}(G)$.
2: $\texttt{m} \Leftarrow \theta \cdot \text{DCOST}(f)$.
3: $\texttt{g1} \Leftarrow \mathcal{M}(G + P_m)$.
4: $\texttt{g2} \Leftarrow \mathcal{SA}(g_1)$.
5: $\texttt{g} \Leftarrow \texttt{g2}$ restricted to G.
6: Output g.

Let $x = \text{DCOST}(G)$ and $y = \text{CCOST}(g_2)$. We first show that the output $\texttt{g}$ is a legal DOLA arrangement. To show that we only need to show that there are no backward edges of G in the arrangement g_2.

Suppose to the contrary, and let $e = (u, v)$ be a backward edge in the arrangement g_2. Then the latency of $e > m$. Hence $y = \text{CCOST}(g_2) > 2m$ ($\geq m$ for P_m and $> m$ for e). On the other hand, we know $y \leq \text{CCOST}(g_1) \leq (1+\delta)(x+m)$. Hence $2m < (1+\delta)(x+m)$, or $m < \theta x$. But $m = \theta \cdot \text{DCOST}(f) \geq \theta x$. This contradicts our assumption that e was a backward edge. Hence $\texttt{g2}$ has only forward edges. Therefore $\texttt{g}$ is a legal DOLA arrangement of G.

We now prove the approximation guarantee. Since $m \geq x$, we have $\text{CCOST}(G + P_m) = x + m$ using Corollary 21. Since $\mathcal{M}$ guarantees $(1+\delta)$-approximation, and the straightening algorithm does not increase the cost,

$$x + m \leq y \leq \text{CCOST}(g_1) \leq (1+\delta)(x+m).$$

Therefore,

$$x \leq y - m \leq (1+\delta)x + \delta m.$$

From the structure of $\texttt{g2}$, and the fact that $\texttt{g}$ is a legal DOLA arrangement, we know $\text{DCOST}(\texttt{g}) = y - m$. But $m = \theta \cdot \text{DCOST}(f) \leq \theta\alpha x$. Therefore,

$$x \leq \text{DCOST}(\texttt{g}) \leq (1+\delta)x + \delta\theta\alpha x = \beta \cdot \text{DCOST}(G).$$

□

The coefficient of α in β is $\theta\delta$. So if $\theta\delta \geq 1$, the above result doesn't say much. The condition $\theta\delta < 1$ translates to $\delta < \sqrt{2} - 1$. This leads to the following theorem.

Theorem 30 (Bootstrapping) *Suppose that un-weighted DCA has a polynomial time $(1 + \delta)$-approximation algorithm for some $\delta < \sqrt{2} - 1$. Put $\Gamma(\delta) = 1 + \frac{2\delta}{1-2\delta-\delta^2}$. Then for every $\epsilon > 0$, there is a polynomial time $(\Gamma(\delta)+\epsilon)$-approximation algorithm for un-weighted DOLA.*

PROOF. Fix a $\delta < \sqrt{2} - 1$. Define $\mu(\alpha) := (1 + \delta)\left[1 + \frac{\alpha\delta}{1-\delta}\right] = a + b\alpha$, for suitable $a, b > 0$. Since $\delta < \sqrt{2} - 1$, $b < 1$. From Proposition 29, we know if there is an α-approximation algorithm for un-weighted DOLA, then there is a $\mu(\alpha)$-approximation algorithm for un-weighted DOLA, thus allowing us to generate a better approximation algorithm. Applying this result $h(n)$-times, we can convert an α-approximation algorithm to a $\mu^{h(n)}(\alpha)$-approximation algorithm. Here $\mu^{h(n)}$ denotes the composition of μ, $h(n)$-times. If $h(n) = n^{O(1)}$ the resulting algorithm also runs in polynomial time.

$\mu^t(\alpha) = a(1+b+\cdots+b^{t-1})+b^t\alpha \leq a/(1-b)+b^t\alpha$, since $b < 1$. Start with the trivial approximation algorithm for DOLA (i.e. return any topological sort of G) which has an approximation ratio of n. Take $h(n) > \log_{1/b}(n/\epsilon)$ so that $b^{h(n)}n < \epsilon$. After $h(n)$ applications of Proposition 29, we get an approximation algorithm with ratio $< \Gamma(\delta) + \epsilon$. Note that $\Gamma(\delta) = a/(1 - b)$ is the fixed point of the map $\alpha \mapsto \mu(\alpha)$. $\square$

As a corollary we have,

Corollary 31 *If un-weighted DCA has a $7/5$-approximation algorithm, then un-weighted DOLA has a 21-approximation algorithm.*

If we are only interested in approximating the cost of the optimal arrangement (not interested in finding a legal arrangement with near optimal cost), then we can do better. Note that we do not expect to find a near optimal arrangement in this case.

Definition 32 *An minimization problem Π is said to have an α-cost-approximation algorithm if there is an polynomial time algorithm $\mathcal{A}$ which given an instance I of Π, returns a* **number** *$\mathcal{A}(I)$ such that $\Pi(I) \leq \mathcal{A}(I) \leq \alpha\Pi(I)$.*

Proposition 33 *If un-weighted DOLA has an α-cost-approximation algorithm and un-weighted DCA has a $(1+\delta)$-cost-approximation algorithm. Then un-weighted DOLA has a $(1 + \delta(1 + \alpha))$-cost-approximation algorithm.*

PROOF. Let $\mathcal{M}$ denote the $(1+\delta)$-cost-approximation algorithm for un-weighted DCA, and $\mathcal{D}$ denote the α-cost-approximation algorithm for un-weighted DOLA.

Require: Input un-weighted directed graph G.
Ensure: z is a $(1+\delta(1+\alpha))$-approximation to DCOST(G).
1: $\mathtt{m} \Leftarrow \mathcal{D}(G)$.
2: $\mathtt{y} \Leftarrow \mathcal{M}(G+P_m)$.
3: Return $z = y - m$.

Let $x = \text{DCOST}(G)$. Then, by Corollary 21, $\text{CCOST}(G+P_m) = x+m$. Since $\mathcal{M}$ guarantees $(1+\delta)$-cost-approximation, $x+m \leq y \leq (1+\delta)(x+m)$. But $m \leq \alpha x$. Therefore, we have $x \leq y-m \leq (1+\delta+\delta\alpha)x$. Hence the result. □

Applying the same boot strapping technique as in Theorem 30, we get

Proposition 34 *Suppose there is a $(1+\delta)$-cost-approximation algorithm for un-weighted DCA for some $\delta < 1$. Then for every $\epsilon > 0$, there is a $(\frac{1+\delta}{1-\delta}+\epsilon)$-cost-approximation algorithm for un-weighted DOLA.*

Corollary 35 *If un-weighted DCA has a $7/5$-cost-approximation algorithm, then un-weighted DOLA has a $7/3$-cost-approximation algorithm.*

We conclude this section with conditional hardness results.

Corollary 36
- *For every $1 \leq c < \sqrt{2}$, there exists $d > 1$ such that it is NP-hard to approximate un-weighted DCA to within c if it is NP-hard to approximate un-weighted DOLA to within d.*
- *For every $1 \leq c < 2$, there exists $d > 1$ such that it is NP-hard to cost-approximate un-weighted DCA to within c if it is NP-hard to cost-approximate un-weighted DOLA to within d.*

PROOF. Follows from Theorem 30 and Proposition 34. □

8 Issue of Connectedness

In this section we show how to reduce DCA instances on arbitrary graphs to DCA instances on weakly connected graphs. Fix a weighted graph G with k weakly connected components $\{G_i\}_{i=1}^k$. Let $n = |V(G)|$ and $n_i = |V(G_i)|$. From Proposition 20, we know that there is an optimal arrangement g for G which is a concatenation of arrangements (not necessarily optimal) g_i of G_i. We start with the observation that g_i depends only on G_i and $n = |V(G)|$.

Then we show how to synthesize weakly connected instances whose solution will allow us to determine g_i. We start with the following

Definition 37 *Given graphs $H_1, \dots, H_k$, define their circular join, denoted $\circ(H_1, \dots, H_k)$ as the graph H, where*

- $V(H) = \cup_i \{V(H_i) \cup \{x_i\}\}$ *where x_i are vertices not appearing in any of the H_i.*
- $E(H) = \cup_i \{E(H_i) \cup \{(v, x_i), (x_{i-1}, v)\}_{v \in V(H_i)}\}$ *where $x_0 = x_k$.*

In other words, place the graphs H_i in that order around a circle, and place new vertices x_i after H_i. Then connect all vertices of H_i to x_i and x_i to all vertices of H_{i+1}. All new edges have unit weight.

Proposition 20 can be extended to circular join's as follows.

Proposition 38 *Let H be the circular join of $H_1, \dots, H_k$, and f any arrangement of H. Then we can construct an arrangement h of H such that*

- $\mathrm{CCOST}(h) \leq \mathrm{CCOST}(f)$,
- $h = h_1 x_1 \dots h_i x_i \dots h_k x_k$, *where h_i is an arrangement of H_i, and*
- $h_i = f|_{H_i}$, *is the restriction of f to H_i.*

PROOF. Let E' denote the edges of H incident to some x_i, and let E'' denote the remaining edges of H. Note that $X = \{x_i\}_{i=1}^k$ is an independent set. Let $H' = (V(H), E')$ and $H'' = (V(H), E'')$. Let f' denote the same arrangement as f, but for the graph H'. Similarly for f''. Let $\alpha := \sum_i (X^+(x_i) + X^-(x_i))$ (see Definition 7).

Now the proof of Corollary 12 actually shows that $\mathrm{CCOST}(f) \geq \mathrm{CCOST}(f') + \alpha$.

Now apply Proposition 20 to the graph H' and the arrangement f', to get g'. g' consists of h_i's ($h_i = f'|_{H_i} = f|_{H_i}$) and x_i's in some order. Rearranging the h_i and x_i's does not change the cost of any edge. Thus we may assume $g' = h_1 x_1 \dots h_i x_i \dots h_k x_k$. Moreover $\mathrm{CCOST}(g') \leq \mathrm{CCOST}(f')$.

Now note that h is the same as the arrangement g' but for the graph H. More over the h-cost for the edges E'' equals α. Thus we have

$$\begin{aligned} \mathrm{CCOST}(f) &\geq \mathrm{CCOST}(f') + \alpha \\ &\geq \mathrm{CCOST}(g') + \alpha \\ &= \mathrm{CCOST}(h) \end{aligned}$$

Hence the result. □

Now for $G = H_1 + \cdots + H_k$, consider $G' = G + P_{k-1}$ and $G'' = \circ(H_1 \ldots H_k)$. Since both G' and G'' have the same number of vertices, we see that the optimal arrangement (as given by Proposition 38 and Proposition 20) are essentially the same. Since G'' is a weakly connected graph (assuming that H_i are the components of G), we have reduced the problem of finding the optimal arrangement for G' to that of finding it for G''. This solves the case of graphs which have a (sufficiently long) path as a component. For the general case we have

Proposition 39 *The problem of finding the minimal circular arrangement for arbitrary weighted graphs (with k components) can be reduced to weakly connected instances if at least two components have size > 2.*

PROOF. Let $G = G_1 + \cdots + G_k$ be the decomposition into weakly connected components. Let $n_i = |V(G_i)|$. Let $g = g_1 \ldots g_k$ be the optimal arrangement given by Proposition 20.

For $1 \leq i \leq k$ consider $\circ(G_i, P_{m_i})$ where $m_i = n - n_i - 3$ (≥ 0 by assumption). Solving this weakly connected instance gives g_i by the discussion above. Thus we have reduced almost all instances to k weakly connected instances.

In some cases, we can reduce the number of weakly connected instances. For example, if $k < \sqrt{n}$, we may assume (by renumbering the components if needed) that $n_k > \sqrt{n}$. Thus we can get $g_1, \ldots, g_{k-1}$ by considering the circular join of $G_1, \ldots, G_{k-1}, P_\ell$ where $\ell = n_k - k - 1$ (so that the circular join has n vertices), and we can get g_k as before. □

If any component has size ≤ 2, then the arrangement of that component in the optimal arrangement is the obvious one. Secondly, having found the arrangement (in the optimal arrangement) of certain components, we may replace those components with paths of the appropriate length, without affecting the arrangement (in the optimal arrangement) of the other components. These two observations, can be used to remove the assumption in Proposition 39.

9 Polynomial Time Approximation Scheme

We conclude this paper with a PTAS for un-weighted DCA on dense graphs. Arora, Frieze and Kaplan [AFK96] give a PTAS for OLA on dense graphs. We show how the same algorithm with minor modifications works for DCA as well. The algorithm gives an arrangement which is at most ηn^3 away from the optimal solution. If the graph is dense, then Proposition 9 shows that the optimum value is $\Omega(n^3)$.

Definition 40 *For constant t, let $I_1, \ldots, I_t$ be a partition of $[n] := \{1, \ldots, n\}$ into consecutive equal sized intervals, such that $I_i = \{it, \ldots, (i+1)t - 1\}$. A **placement** is a mapping from the vertex set to the set $\{I_1, \ldots, I_t\}$. A placement f' is **proper** if $|f'^{-1}(I_i)| = |I_i|$ for each i. Given any mapping $f : V \mapsto [n]$, we denote by f' the induced placement. The cost of a placement f', denoted by $\mathrm{CP}(f')$ is defined to be $\sum_{(u,v)\in E}(f'(v) - f'(u) \mod t)$.*

Proposition 41 *If f is any arrangement, $|\,\mathrm{CCOST}(f) - \mathrm{CP}(f')n/t| \leq n^3/t$.*

PROOF. Consider any edge (u, v) with latency $x = f(v) - f(u) \mod n$.

- If $f'(u) \neq f'(v)$, then $f'(v) - f'(u) \mod t$ is $\lfloor xt/n \rfloor$ or $\lfloor xt/n \rfloor + 1$. Hence this edge contributes a difference of at most n/t. If $f'(u) = f'(v)$ and $f(u) < f(v)$, then this edge contributes $f(v) - f(u) < n/t$ to the difference.
- If $f'(u) = f'(v)$ and $f(v) < f(u)$, then this edge contributes $n - (f(u) - f(v)) \leq n - 1$ to the difference. If (u, v) is an edge of this kind then $x > n - n/t$. Each interval can contain a maximum of $\binom{n/t}{2} < n^2/2t^2$ such edges. Hence the total number of edges of this kind $< (n^2/2t^2) \cdot t = n^2/2t$.

Thus an upper bound to the difference is $|E|n/t$ for edges of the first kind and $(n^2/2t) \cdot n$ for edges of the second kind, making a total of n^3/t. □

Proposition 42 *Let f and g are arrangements such that $|\,\mathrm{CP}(f') - \mathrm{CP}(g')| \leq \epsilon n^2$, then $|\,\mathrm{CCOST}(f) - \mathrm{CCOST}(g)| \leq (2 + \epsilon)n^3/t$.*

PROOF. From hypothesis and Proposition 41 and triangle inequality, we have

$$
\begin{aligned}
&|\,\mathrm{CCOST}(f) - \mathrm{CCOST}(g)| \\
\leq &|\,\mathrm{CCOST}(f) - \frac{n}{t}\,\mathrm{CP}(f')| + |\frac{n}{t}\,\mathrm{CP}(f') - \mathrm{CCOST}(g)| \\
\leq &\frac{n^3}{t} + \frac{n}{t}|\,\mathrm{CP}(f') - \mathrm{CP}(g')| + |\frac{n}{t}\,\mathrm{CP}(g') - \mathrm{CCOST}(g)| \\
\leq &\frac{n^3}{t} + \frac{\epsilon n^3}{t} + \frac{n^3}{t} \\
\leq &(2 + \epsilon)\frac{n^3}{t}
\end{aligned}
$$

□

Let $c > 0$ be a large constant, and $\eta > 0$ the desired degree of approximation. Set $t = c/\eta$ and $\epsilon = c - 2$. Now Proposition 42 shows that in order to find an arrangement within an additive error of ηn^3 of the optimal arrangement F, it is enough to find a placement within an additive error of ϵn^2 from F'. Once we find such a placement, we output any arrangement which maps to this placement.

For the rest of this section, we go about finding a placement within an additive error of ϵn^2 from the optimal placement (and hence from F' as well).

Notation Letters $f, g, \ldots$ will denote arrangements, $f', g', h', \ldots$ denote placements and $\hat{f}, \hat{g}, \ldots$ denote maps from $S \mapsto [t] := \{1, \ldots, t\}$ (to be defined very soon). If more than one of $f, f', \hat{f}$ (or $g, g', \hat{g} \ldots$) is defined then they correspond to each other, via the obvious correspondences. Finally $\tilde{}$ is used to denote a fractional quantity.

9.1 Finding a good placement

Let g' be an optimal placement (i.e. placement corresponding to some optimal arrangement). The contribution of vertex u to the cost is

$$\mathrm{CP}_u(g') := \sum_{(u,v)\in E} (g'(v) - g'(u) \mod t)$$

In order to estimate the above term, we sample a multi-set S of size $O(\log n/\delta^2)$, where $\delta = \bar{\epsilon}/t$ and $\bar{\epsilon} = \epsilon/3$. Having sampled an S, we enumerate all functions $\hat{h} : S \mapsto \{1, \ldots, t\}$, and for each such $\hat{h}$ we find an approximation h' to the best placement whose restriction to S is $\hat{h}$. Let g' be the placement which has the least cost among all h''s. Then w.h.p over choice of S, g' is the placement we are looking for.

9.1.1 Going from $\hat{h}$ to h'

Given a function $\hat{h} : S \mapsto \{1, \ldots, t\}$, we construct a linear program $M_{\hat{h}}$ as follows:

For each vertex u and interval I_k, we compute an estimate e_{uk} of the cost of assigning vertex u to interval I_k in any arrangement g for which $\hat{g} = \hat{h}$:

$$e_{uk} = \frac{n}{|S|} \sum_{(u,v)\in E, v\in S} (\hat{h}(v) - k \mod t)$$

This is well defined even when $u \in S$. More over if v appears many times in S, it contributes that many times to the sum. Now we find an optimum fractional placement $\tilde{h}'$ for which the cost of assigning vertex u to interval I_k is almost e_{uk}, i.e. within $[e_{uk} \pm \bar{\epsilon} n]$ by solving the following linear program $LP(\hat{h})$:

$$\text{minimize } \sum_{u \in V} \sum_{k \in [t]} e_{uk} x_{uk}$$

s.t.

$$\begin{array}{rcll} \sum_{u \in V} x_{uk} & = & n/t & \forall k \in [t] \\ \sum_{k \in [t]} x_{uk} & = & 1 & \forall u \in V \\ \sum_{(u,v) \in E} \sum_{j \in [t], j \neq k} (j - k \mod t) x_{vj} & \leq & e_{uk} + \bar{\epsilon} n & \forall u \in V, k \in [t] \\ \sum_{(u,v) \in E} \sum_{j \in [t], j \neq k} (j - k \mod t) x_{vj} & \geq & e_{uk} - \bar{\epsilon} n & \forall u \in V, k \in [t] \\ 0 \leq & x_{uk} & \leq 1 & \forall u \in V, k \in [t] \end{array}$$

For a fixed $\hat{h}$, let y be the optimal fractional placement returned by the linear program. We then construct a placement r by setting $r(u) = k$ with probability y_{uk}. This placement may not be proper (i.e. some interval may have more than n/t vertices). Now construct a proper placement h' by arbitrarily moving vertices from over populated intervals to under populated intervals.

9.2 Proof of correctness

[The proof of correctness is the same as in [AFK96] with the obvious modifications. So, we just state the lemmas, and leave the proofs to the reader.]

It is possible that some of the linear programs above are infeasible. We now show that when $\hat{h}$ corresponds to the optimal placement, then $LP(\hat{h})$ is feasible. More specifically g' is a feasible solution to $LP(\hat{g})$, follows from

Lemma 43 (Lemma 4.2 of [AFK96]) *Let f' be any placement. Let S be a randomly chosen multi-set of $O(\log n/\delta^2)$-vertices, and $\hat{f} = f'|_S$. Then whp over choice of S,*

$$\left| e_{uk} - \sum_{(u,v) \in E} (f'(v) - k \mod t) \right| \leq \delta n t$$

for any $u \in V, k \in [t]$.

Let $\tilde{g}$ be the optimal fractional placement for the linear program $LP(\hat{g})$. If

$\phi(\cdot)$ denote the value of the objective function of $LP(\hat{g})$, we have

$$\phi(\tilde{g}) \leq \phi(g') \leq \mathrm{CP}(g') + \bar{\epsilon}n^2$$

since g' is a feasible solution which satisfies upper bound estimates for e_{uk}.

Lemma 44 (Lemma 4.3 of [AFK96]) *Let r' be the proper placement constructed from $\tilde{g}$. Then $\mathrm{CP}(r') \leq \phi(\tilde{g}) + \bar{\epsilon}n^2 + o(n^2)$.*

Now taking $\bar{\epsilon} = \epsilon/3$, we have that $\mathrm{CP}(r') \leq \mathrm{CP}(g') + \epsilon n^2$. The running time of this algorithm as stated is $n^{O(\log n/\eta^2)}$. [AFK96] derandomizes this algorithm using constant degree expanders to run in time $n^{O(1/\eta^2)}$.

10 Discussion

Motivated by a problem related to design of cyclic multicast schedule, we studied DCA problem in this paper. Considering current trend in technologies and applications, cyclic multicast that pays heed to data dependencies should play very pivotal role in future [CLP01].

Our research pointed our certain negative aspects of DCA problem, namely, it does not have a polynomial time algorithm and it does not even admit a polynomial time approximation scheme for arbitrary graph instance unless P=NP. Yet it is possible that DCA problem might be tenable if restricted to certain special kinds of graphs that have practical significance. Literature has many such instances of polynomial time algorithms for OLA problem, *e.g.*, unweighted trees [Shi79,Chu84], outer planar graphs [FH88], wheels, complete bipartite graphs [JM92], etc. Can one hope for same in case of DCA? Or is it also *too hard*?

Assuming DOLA to be non-approximable within any constant factor, we could show a lower bound of $\sqrt{2} - \epsilon$ for DCA approximation. We believe it to be far from being tight. In fact, there is a conspicuous lack of hardness of approximation results even for OLA and DOLA. They stand as natural open problems.

Liberatore provides few heuristics [Lib02b,Lib02a] to solve the DCA problem. Naor and Schwartz [SN04] give an $\widetilde{O}(\log n)$-approximation algorithm for DCA. However this approximation algorithm while theoretically good, is impractical. Therefore it is an interesting open question to design an efficient approximation algorithm for DCA problem.

References

[AAF98] K C Almeroth, M H Ammar, and Z Fei. Scalable delivery of web pages using cyclic best-effort multicast. In *IEEE INFOCOM*, pages 1214–1221, March 1998.

[Ach98] S Acharya. *Broadcast Disks: Dissemination-based Data Management for Assymetric Communication Environments.* PhD thesis, Brown University, May 1998.

[AFK96] S Arora, A Frieze, and H Kaplan. A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. In *37th Annual IEEE Symposium on Foundations of Computer Science*, pages 21–30, October 1996.

[AH73] D Adolphson and T C Hu. Optimal linear ordering. *SIAM Journal on Applied Mathematics*, 25(3):403–423, 1973.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, May 1998.

[Chu81] F R K Chung. *Theory and Applications of Graphs*, chapter Some Problems and Results in Labelings of Graphs, pages 255–264. John Wiley & Sons, New York, 1981.

[Chu84] F R K Chung. On optimal linear arrangements of trees. *Comp. & Maths with Applications*, 10(1):43–60, 1984.

[CLP01] P K Chrysanthis, V Liberatore, and K Pruhs. Middleware support for multicast-based data dissemination: A working reality, 2001. White paper.

[ES75] S Even and Y Shiloach. NP-Completeness of several arrangement problems. Technical Report 43, Isreal Institute of Technology, 1975.

[FH88] G N Frederickson and S E Hambrusch. Planar linear arrangements of outerplanar graphs. *IEEE Transactions on Circuits and Systems*, 35(3):323–332, 1988.

[FK99] A M Frieze and R Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.

[GJS76] M R Garey, D S Johnson, and L Stockmeyer. Some simplified NP-Complete graph problems. *Theoretical Computer Science*, 3(1):237–267, 1976.

[HGLW87] G Herman, G Gopal, K C Lee, and A Weinrib. The datacycle architecture for very high throughput data systems. In *ACM SIGMOD International Conference on Management of Data*, pages 97–103, May 1987.

[JM92] M Juvan and B Mohar. Optimal linear labelings and eigenvalues of graphs. *Discrete Applied Mathematics*, 36:153–168, 1992.

[KMSV98] Sanjeev Khanna, Rajeev Motwani, Madhu Sudan, and Umesh V. Vazirani. On syntactic versus computational views of approximability. *SIAM J. Comput.*, 28(1):164–191, 1998.

[KR01] B Krishnamurthy and J Rexford. *Web Protocols and Practice*. Addison-Wesley, Boston, 2001.

[Lib02a] V Liberatore. Circular arrangements. In *ICALP*, pages 1054–1066, 2002.

[Lib02b] V Liberatore. Multicast scheduling for list requests. In *IEEE INFOCOM*, pages 1129–1137, June 2002.

[PY91] C H Papadimitriou and M Yannakakis. Optimmization, approximation and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[Rab01] M Rabinovich. Resource management issues in content delivery networks (CDNs). In *DIMACS Workshop on Resource Management and Scheduling in Next Generation Networks*, 2001.

[RR98] S Rao and A W Richa. New approximation techniques for some ordering problems. In *9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 211–218. ACM-SIAM, January 1998.

[Shi79] Y Shiloach. A minimum linear arrangement algorithm for undirected trees. *SIAM Journal of Computing*, 8(1):15–32, 1979.

[SN04] Roy Schwartz and Joseph (Seffi) Naor. The directed circular arrangement problem. In *Symposium on Discrete Algorithm*. ACM SIAM, 2004.

[SNSR99] J Shanmugasundaram, A Nithrakashyap, R Sivasankaran, and K Ramamritham. Efficient concurrency control for broadcast environments. In *ACM SIGMOD International Conference on Management of Data*, pages 85–96, June 1999.

[Ste98] W R Stevens. *Unix Network Programming*. PTR PH, 1998.