Short Tours through Large Linear Forests

Uriel Feige[1]*, R. Ravi[2]**, and Mohit Singh[3]

[1] Department of Computer Science, The Weizmann Institute, Rehovot, Israel. Email: uriel.feige@weizmann.ac.il.
[2] Tepper School of Business, Carnegie Mellon University. Email: ravi@cmu.edu
[3] Microsoft Research, Redmond, USA. Email:mohits@microsoft.com

Abstract. A tour in a graph is a connected walk that visits every vertex at least once, and returns to the starting vertex. Vishnoi [18] proved that every connected d-regular graph with n vertices has a tour of length at most (1 + o(1))n, where the o(1) term (slowly) tends to 0 as d grows. His proof is based on van-der-Warden's conjecture (proved independently by Egorychev [8] and by Falikman [9]) regarding the permanent of doubly stochastic matrices. We provide an exponential improvement in the rate of decrease of the o(1) term (thus increasing the range of d for which the upper bound on the tour length is nontrivial). Our proof does not use the van-der-Warden conjecture, and instead is related to the linear arboricity conjecture of Akiyama, Exoo and Harary [1], or alternatively, to a conjecture of Magnant and Martin [12] regarding the path cover number of regular graphs. More generally, for arbitrary connected graphs, our techniques provide an upper bound on the minimum tour length, expressed as a function of their maximum, average, and minimum degrees. Our bound is best possible up to a term that tends to 0 as the minimum degree grows.

1 Introduction

A *tour* in a graph is a connected walk that starts at a vertex, visits every vertex of the graph at least once, and returns to the starting vertex. The *length* of the tour is the number of steps of the corresponding walk. Vishnoi [18] proved the following theorem.

Theorem 1. *[18] Every n-vertex d-regular connected graph has a tour of length at most $\left(1+\sqrt{\frac{64}{\log d}}\right)n$. Moreover, there is a randomized polynomial time algorithm that with high probability finds such a tour.*

The existential part of the proof of Theorem 1 is based on van-der-Warden's conjecture (proved independently by Egorychev [8] and by Falikman [9]) regarding the permanent of doubly stochastic matrices. (See also Section 3 of [2] for related results.) The algorithmic part is based on randomized algorithms for approximating the permanent [11]. We provide the following strengthening of Theorem 1.

* Work supported in part by the Israel Science Foundation (grant No. 621/12) and by the I-CORE Program of the Planning and Budgeting Committee and The Israel Science Foundation (grant No. 4/11).

** Work supported in part by NSF grant CCF-1218382.

Theorem 2. *Every n-vertex d-regular connected graph has a tour of length at most $\left(1+O\left(\frac{1}{\sqrt{d}}\right)\right)n$. Moreover, there is a randomized polynomial time algorithm that finds such a tour.*

Our proof does not use the van-der-Warden conjecture, and instead works by constructing a large *linear forest*. A linear forest is an acyclic subgraph where the degree of any vertex is at most two. Equivalently, a linear forest is a vertex disjoint union of paths. We prove the following theorem about existence of large linear forests in d-regular graphs.

Theorem 3. *Every n-vertex d-regular graph has a linear forest of size $\left(1-O\left(\frac{1}{\sqrt{d}}\right)\right)n$, and moreover, such a linear forest can be found in randomized polynomial time.*

We observe that certain unproved conjectures (specifically, the linear arboricity conjecture of [1], or alternatively, a conjecture of [12] regarding the path cover number of regular graphs) would imply that every d-regular graph has a tour of length at most $\left(1+O\left(\frac{1}{d}\right)\right)n$. This bound would be best possible up to the hidden constant in the O notation, as there are d-regular graphs in which every tour is of length $(1+\Omega(\frac{1}{d}))n$. See more details in Section 4.

A linear forest with a large size given by Theorem 3 is useful for constructing a spanning tree with few odd-degree nodes: indeed, extending the forest to any spanning tree introduces odd degree nodes of the same order as the number of components of the forest. For an even cardinality set T of vertices, a T-join is a collection of edges which has odd-degree exactly on vertices in T. Following Christofides' algorithm [6], it then suffices to construct a T-join on these few odd degree nodes. We show that, in a graph whose minimum degree is large, there is always a small size T-join when $|T|$ is small in the following theorem.

Theorem 4. *Let $G(V,E)$ be an arbitrary connected graph with n vertices and minimum degree δ, and let $T \subset V$ be an arbitrary set of vertices of even cardinality. Then there is a T-join with fewer than $2|T|+\frac{3n}{\delta+1}$ edges.*

The above theorem then, along with Theorem 3, directly gives Theorem 2. We also observe that Theorem 4 can be thought of as a generalization of the classical result [14], up to additive constant terms, that every graph with minimum degree δ has diameter at most $\frac{3n}{\delta+1}$. This follows since when $T=\{u,v\}$ then the smallest T-join is exactly the shortest path between u and v.

In contrast to the result of Vishnoi [18], our results extend naturally to nearly regular graphs and we prove the following general theorem.

Theorem 5. *Let G be a connected n-vertex graph with maximum degree Δ, average degree d, and minimum degree δ. Then G has a tour of length at most*

$$\left(1+\frac{\Delta-d}{\Delta}+O\left(\frac{1}{\sqrt{\Delta}}\right)+O\left(\frac{1}{\delta}\right)\right)n.$$

Moreover, there is a randomized polynomial time algorithm that finds such a tour.

Theorem 5 provides tours not much larger than n when Δ is close to d, and δ is fairly large. The term $\frac{\Delta-d}{\Delta}$ is best possible, but the error terms $O\left(\frac{1}{\sqrt{\Delta}}\right) + O\left(\frac{1}{\delta}\right)$ can possibly be improved. See more details in Section 3.1.

1.1 Related Work

There has been extensive recent work on approximation algorithms for the graph-TSP problem, which is the same as that of finding a minimum length tour of a given undirected graph. While Christofides' algorithm [6] gives a $\frac{3}{2}$ approximation even for graph-TSP, a small but constant improvement was presented by Oveis-Gharan et al. [16]. Mömke and Svensson [13] improved this significantly while further improvements by Sebö and Vygen [17] have brought the current best approximation factor for graph-TSP down to $\frac{7}{5}$. The methods of Mömke and Svensson [13] also give a $\frac{4}{3}$ approximation algorithm for subcubic 2-connected graphs.

Another line of work has focused on graph theoretic methods to obtain improved approximation factors: Boyd et al. [5] showed a $\frac{4}{3}$ approximation for 2-connected cubic graphs while Correa et al [7] gave an algorithm that finds a tour of length at most $(\frac{4}{3} - \frac{1}{61236})n$ in n-node 2-connected cubic graphs. For general, d-regular connected graphs, Vishnoi [18] gave an algorithm for finding tours of length at most $(1+\frac{8}{\sqrt{\log d}})n$.

2 Small T-joins in Regular Graphs

In this section we prove Theorem 4 which follows directly from the following strengthening.

Theorem 6. *Let $G(V, E)$ be an arbitrary connected graph with n vertices and minimum degree δ, and let $T \subseteq V$ be an arbitrary set of vertices of even cardinality. Then there is a T-join with fewer than $2|T| + \frac{3n}{\delta+1} - 2\nu$ edges, where ν is the number of connected components in the T-join. Moreover, such a T-join can be found in polynomial time.*

Proof. Given $u, v \in V$, let $d(u, v)$ denote the number of edges along the shortest path between u and v in G. Consider the following iterative procedure for constructing a set $S \subset V$ together with a set P of *virtual edges*, each of length 3. Initially, place an arbitrary vertex v in S. Thereafter, in every iteration, consider an arbitrary vertex (say, u) whose distance from S is exactly 3. If there is no such vertex the procedure ends. Given such a vertex u, let w be an arbitrary vertex in S with $d(w, u) = 3$. Add u to S and the virtual edge (w, u) to P. This completes the description of the iteration.

Observe that necessarily $|S| \leq \frac{n}{\delta+1}$, because every vertex of S excludes all its neighbors from being in S, and the neighborhoods of vertices in S are disjoint. Observe also that the graph $G'(S, P)$ induced on S and the virtual edges is a tree.

Associate with every vertex $v \in T \setminus S$ the vertex $u \in S$ that is closest to v (breaking ties arbitrarily), and observe that $d(u, v) \leq 2$ (due to the maximality of S). Add an auxiliary edge (u, v) to P, with length $d(u, v)$. Consider now the tree T' whose vertices are $T \cup S$, and whose edge set is P. The total number of virtual edges in T' is exactly

$|S \cup T| - 1$, exactly $|S| - 1$ of these virtual edges have length 3, and the remaining edges have length at most 2. Within T', find the unique T-join (where a tree edge is in the T-join iff each of the two connected components that are formed by removing it has an odd number of vertices from T). Let ν' denote the number of connected components (with respect to T') in this T-join. Then the number of virtual edges in the T-join is exactly $|S \cup T| - \nu'$, and their total length is at most $3(|S| - 1) + 2|T \setminus S| - 2(\nu' - 1) < 3|S| + 2|T| - 2\nu'$.

Now replace the virtual edges of the T-join by edges along the corresponding shortest paths in G. The total number of edges needed is less than $3|S| + 2|T| - 2\nu'$. In the process of replacing virtual edges by paths, the same edge of G might be introduced multiple times. If so, any double occurrence of an edge is removed (as this does not change the parity of degrees), so as to make the resulting T-join a simple subgraph of G with no parallel edges. The removal of a set of edges parallel to each other might add 1 to the number of connected components, but decreases the number of edges in the T-join by at least 2. Hence if the final number of connected components in the T-join is ν, then the total number of edges in the T-join is less than $3|S| + 2|T| - 2\nu \leq 2|T| + \frac{3n}{\delta+1} - 2\nu$, as desired.

We now prove the following corollary.

Corollary 1. *Let G be a connected graph with n vertices and minimum degree δ, and let F be a linear forest in G. Then given F, one can find in polynomial time a tour of G of length smaller than $2n - |F| + \frac{5n}{\delta+1}$.*

Proof. Without loss of generality, assume that F is a maximal linear forest. This implies that isolated vertices cannot be neighbors of each other or neighbors of endpoints of paths, and endpoints of different paths cannot be neighbors of each other. The forest F induces in G exactly $n - |F|$ connected components, where a connected component is either a path or an isolated vertex.

We first describe a process for adding edges from G to the forest so that it becomes connected. In the process we may add the same edge more than once, and hence we shall obtain a connected spanning multigraph. The governing consideration in deciding which edges to add is that of keeping the number of odd degree vertices as small as possible (in particular, all odd degree vertices will be of degree one). The rules for adding edges are as follows:

1. If a component is an isolated vertex v, add an arbitrary edge incident to v (hence v joins some other connected component), and double this edge. Hence the number of connected components drops by one, the number of edges grows by two, and the number of odd degree vertices does not change.
2. If there are two vertices u and v of degree one in different connected components C_u and C_v with $d(u, v) = 2$ then connect them by a shortest path. Hence the number of connected components drops by one, the number of edges grows by two, and the number of odd degree vertices drops by two. Observe that the path between u and v might go through another component C', in which case the number of connected components should have dropped by two. However, for uniformity of the analysis (and without affecting its correctness) we shall count C' as a component distinct from the new component formed by C_u and C_v.

When none of the above two rules applies, let q denote the number of remaining connected components (each of which has two vertices of degree one).

The number of edges (including parallel edges) added by the above procedure is exactly $2(n - |F| - q)$.

Observe that if we take one vertex of degree one from each remaining component, no two such vertices share a neighbor (otherwise rule 2 above would apply). Hence $q \leq \frac{n}{\delta+1}$.

Let T denote the set of odd degree vertices that still remain, and note that $|T| = 2q \leq \frac{2n}{\delta+1}$. Now find a T-join in G using Theorem 6. This T-join has less than $2|T| + \frac{3n}{\delta+1} - 2\nu$ edges, where ν denotes the number of connected components that remain.

The union of the q components and the T-join is a spanning Eulerian subgraph of G with ν connected components. It can be made connected (and kept Eulerian) by adding $\nu - 1$ pairs of parallel edges. Thereafter, an Eulerian cycle can serve as a tour of G. The total number of edges (counting multiplicities) in this union is less than

$$|F| + 2(n - |F| - q) + 4q + \frac{3n}{\delta + 1} - 2\nu + 2(\nu - 1) < 2n - |F| + \frac{5n}{\delta + 1}$$

proving the theorem.

3 Large Linear Forests and Fractional Arboricity

The *fractional linear arboricity* of a graph G is the minimum number of linear forests needed to cover every edge where we are allowed to pick a linear forest fractionally. Given Corollary 1, the proof of Theorem 2 would follow from a lower bound on the size of the maximum linear forest in regular graphs. Indeed, we prove a stronger result and show that fractional linear arboricity of any d-regular graph is at most $\frac{d-O(\sqrt{d})}{2}$.

Theorem 7. *There exists a randomized algorithm that given a d-regular graph $G = (V, E)$ returns a linear forest F such that for each edge $e \in E$, the probability $e \in F$ is at least $\frac{2}{d+O(\sqrt{d})}$. Thus the fractional arboricity of any d-regular graph is at most $\frac{d-O(\sqrt{d})}{2}$.*

Before we prove Theorem 7, we prove Theorem 3.

Proof. Sample a random linear forest F as given by Theorem 7. The expected size of the forest F is at least

$$\sum_{e \in E} Pr[e \in F] \geq |E| \cdot \frac{2}{d + O(\sqrt{d})} \geq \frac{nd}{2} \cdot \frac{2}{d + O(\sqrt{d})} \geq \left(1 - O\left(\frac{1}{\sqrt{d}}\right)\right) n$$

as required.

We can now prove Theorem 2.

Proof. Theorem 3 implies that every n-vertex d-regular graph has a linear forest of size $(1 - O(\sqrt{\frac{1}{d}}))n$, and moreover, such a linear forest can be found in polynomial time. Plugging this value of $|F|$ in Corollary 1 proves the theorem.

Thus it remains to prove Theorem 7.

Proof (Proof of Theorem 7). We shall now describe an iterative algorithm for constructing a random linear forest F. We shall assume for simplicity that n is a power of 2. This assumption has negligible effect on our bounds.

In the beginning of iteration i for $i = 1, 2, \ldots$ we have a directed graph $G_i = (V_i, E_i)$ where the maximum out/in-degree of every vertex is d_i where $d_i \simeq \frac{d}{2^i}$ and there are no parallel arcs (but there can be two anti-parallel arcs). The vertex set of G_i is obtained by identifying vertices of G; thus each vertex of V_i corresponds to subset of vertices in V and these subsets form a partition of V. We also maintain that edges included in F up to iteration $i - 1$ have both their endpoints contracted to the same vertex in V_i. For $i = 1$, we initialize G_1 as follows. If d is even, we pick an Eulerian traversing all edges of G and orient the edges by picking an orientation of the tour and we set $d_1 = \frac{d}{2}$. If d is odd, we first add a matching of auxiliary edges to G. Observe that the multiplicity of any edge is at most two after adding the matching. Now we pick an Eulerian orientation which traverses any two parallel edges right after each other. Now consider the orientation of the edges as given by the Eulerian tour. Clearly, there are no parallel arcs as any edge of multiplicity two is oriented as a pair of anti-parallel arcs. In this case, we set $d_1 = \frac{d+1}{2}$.

In each iteration i, we do the following steps.

1. Pair the vertices of G_i in arbitrary manner. Then form a directed bipartite graph D_i with bipartition $L_i \cup R_i = V_i$, where from each pair of vertices one vertex is included in L_i and the other in R_i, uniformly at random and independently for each pair. Remove all arcs with both endpoints in L_i or both endpoints in R_i to obtain a directed bipartite graph.
2. Next, scan all vertices one by one, and if a vertex has current in- or out-degree more than $d_{i+1} = \lceil \frac{d_i}{2} \rceil$, delete a uniformly random set of in- or out-edges until the degree is exactly d_{i+1}. After this pass, all vertices have in- and out-degree at most d_{i+1} but some may have a strictly smaller degree.
3. Consider the bipartite graph formed by edges directed from L_i to R_i. This bipartite graph has maximum degree d_{i+1}. Add *auxiliary* edges between vertices of L_i and R_i of degree less than d_{i+1} in an arbitrary manner (allowing also parallel edges), until a regular bipartite multi-graph of degree d_{i+1} is obtained.
4. Legally color the edges of this regular bipartite multi-graph with d_{i+1} colors, thus obtaining d_{i+1} perfect matchings. Select uniformly at random one of the color classes as the perfect matching N_i.
5. Let N_i' denote the set of edges which go from R_i to L_i and are anti-parallel to an edge in N_i. Now do one of the following steps.
   (a) Select N_i' as matching M_i with probability $\frac{2}{d_{i+1}}$ and end the algorithm.
   (b) Otherwise, with probability $1 - \frac{2}{d_{i+1}}$, let $M_i = N_i$, and remove all arcs of N_i'. Remove all arcs that go from L_i to R_i and unify the endpoints of M_i. Thus

in the contracted graph, we only retain edges that go from R_i and L_i, and the out/in-degree of each vertex is at most d_{i+1}. Observe that the contracted graph is a simple graph, with no parallel edges and no self loops. This contracted graph serves as G_{i+1} for the next iteration $i+1$.

If step 5(a) is never invoked, the algorithm ends after $\log n$ iterations, as by them the whole graph is contracted to a single vertex. The final output F of the algorithm is the union of all M_i, excluding all the auxiliary edges.

Proposition 1. *The output of the algorithm is a linear forest.*

Proof. Add to the output of the algorithm also all the auxiliary edges that were used in order to extend the M_i matchings into perfect matchings. Considering the graph induced on the edges of all M_i and the auxiliary edges, it can be verified by induction that every vertex in iteration i corresponds to a directed path with exactly 2^i vertices, where the operation performed in iteration i matches such paths in pairs, and concatenates the two members of a pair with a directed (original or auxiliary) edge. Hence with the auxiliary edges the final output of the algorithm is a collection of vertex disjoint paths. Removing the auxiliary edges leaves a vertex disjoint set of paths (some of which might be isolated vertices), which by definition is a linear forest.

Lemma 1. *For any i, an edge e is deleted in Step 2 with probability at most $\frac{8}{\sqrt{d_{i+1}}}$.*

Proof. Let $e=(u,v)$ be any arc $e \in G_i$. The probability that $e \in E(R_i, L_i)$ is at least $\frac{1}{4}$ (exactly $\frac{1}{2}$ if u and v are paired and exactly $\frac{1}{4}$ otherwise), and likewise for $e \in E(L_i, R_i)$. Let us first condition on the event that either $e \in E(R_i, L_i)$ or $e \in E(L_i, R_i)$ and calculate the probability that e is deleted in Step (2) of iteration i. This can happen if the out-degree of u or in-degree of v is more than d_{i+1}. Let us concentrate on the event that e is deleted due to high out-degree at u. For each pair $p=\{w,w'\}$ of vertices in G_i, let X_p denote the indicator random variable for the event that either (u,w) or (u,w') is in $E(R_i,L_i)$. If u has out arcs to both w and w', then $X_p=1$ with probability one. If u has an out-arc to exactly one of w or w', then $X_p=1$ with probability $\frac{1}{2}$. Moreover, these random variables are independent since we make decisions for each pair independently. Thus the out-degree of u is $X=\sum_p X_p$ where $E[X] \le \sum_p E[X_p] \le d_{i+1}$. Since X is a sum of $\{0,1\}$-valued independent Bernoulli random variables, standard Chernoff bounds imply that

$$Pr[deg^{out}(u) \ge d_{i+1} + c\sqrt{d_{i+1}}] \le e^{-\frac{c^2}{3}}.$$

Since we delete a random set of required number of edges at u, we obtain that

$$
\begin{aligned}
& Pr[e \text{ is deleted due to high out-degree at } u] && (1)\\
& \leq \sum_{c=0}^{\infty} \frac{(c+1)\sqrt{d_{i+1}}}{d_i + c\sqrt{d_{i+1}}} Pr[d_{i+1} + c\sqrt{d_{i+1}} \leq deg^{out}(u) \leq d_{i+1} + (c+1)\sqrt{d_{i+1}}] && (2)\\
& \leq \sum_{c=0}^{\infty} \frac{(c+1)\sqrt{d_{i+1}}}{d_i + c\sqrt{d_{i+1}}} \cdot e^{-\frac{c^2}{3}} && (3)\\
& \leq \frac{1}{\sqrt{d_{i+1}}} \sum_{c=0}^{\infty} (c+1) \cdot e^{-\frac{c^2}{3}} \leq \frac{4}{\sqrt{d_{i+1}}} && (4)
\end{aligned}
$$

An identical bound holds for the event that e is deleted due to high in-degree at v. Thus, from the union bound, it follows that

$$Pr[e \text{ is deleted in Step 2}] \leq \frac{8}{\sqrt{d_{i+1}}}.$$

We now lower bound the probability that any e is included in F. We prove the following lemma for the first $T = \frac{\log d}{2}$ iterations. For this lemma, we assume that d is also a power of two for ease of analysis. This assumption has a negligible effect on the bounds.

Lemma 2. *For any* $1 \leq i \leq T$,

$$Pr[e \in F | e \in G_i] \geq \frac{1}{d_i + \left(2^{\frac{i}{2}+1} + 30(\sqrt{2}+1)\right)\sqrt{d_i}} \tag{5}$$

Proof. The proof is by reverse induction on i. For $i = T$, we have $2^{T/2} = d^{1/4}$ and $d_T = \frac{d}{2^T} = \sqrt{d}$. Thus we have that

$$\frac{1}{d_T + \left(2^{\frac{T}{2}+1} + 30(\sqrt{2}+1)\right)\sqrt{d_T}} = \frac{1}{d_T + \left(2\sqrt{d_T} + 30(\sqrt{2}+1)\right)\sqrt{d_T}} \leq \frac{1}{3d_T}. \tag{6}$$

We will show that

$$Pr[e \in F | e \in G_T] \geq \frac{1}{3d_T} \geq \frac{1}{d_T + \left(2^{\frac{T}{2}+1} + 30(\sqrt{2}+1)\right)\sqrt{d_T}}$$

which will prove the base case.

The chance that $e \in E(L_i, R_i)$ is at least $\frac{1}{4}$. From Lemma 1, e is removed in Step 2 with probability at most $\frac{8}{\sqrt{d_{T+1}}}$. Since, each color class is chosen with probability $\frac{1}{d_{T+1}}$, we obtain that

$$Pr[e \in N_T | e \in G_T] \geq \left(\frac{1}{4} - \frac{8}{\sqrt{d_{T+1}}}\right) \cdot \frac{1}{d_{T+1}}$$

But then it is included in M_T with probability $1 - \frac{2}{d_{T+1}}$ independent of the earlier events. Thus

$$Pr[e \in M_T | e \in G_T] \geq \left(\frac{1}{4} - \frac{8}{\sqrt{d_{T+1}}}\right) \frac{1}{d_{T+1}} \cdot \left(1 - \frac{2}{d_{T+1}}\right) \geq \frac{1}{6d_{T+1}} = \frac{1}{3d_T}$$

This proves the base case of the induction.

Now consider any i and let $e \in G_i$. Then e can be included in F in the following three events. Firstly, if it is in $E(L_i, R_i)$, then it can included in N_i and then chosen in M_i. Secondly, if it is in $E(R_i, L_i)$ and if one of the anti-parallel edges to it is chosen in N_i then it is included in N_i' and can be chosen in M_i. Lastly, if it is $E(R_i, L_i)$ but M_i is chosen as N_i and it is not deleted in Step 5, it can be chosen in G_{i+1}. We will calculate the probabilities of the first two events and apply induction to the last event to prove the inductive hypothesis. We have the following inequalities. All events are conditioned on the event that $e \in G_i$.

$$\begin{aligned} Pr[e \in F] =& Pr[e \in E(L_i, R_i)] \cdot Pr[e \in F | e \in E(L_i, R_i)] \\ &+ Pr[e \in E(R_i, L_i)] \cdot Pr[e \in M_i | e \in E(R_i, L_i)] \\ &+ Pr[e \in E(R_i, L_i)] \cdot Pr[e \in G_{i+1} | e \in E(R_i, L_i)] \cdot Pr[e \in F | e \in G_{i+1}] \end{aligned}$$

Now, we calculate each of terms.

$$\begin{aligned} Pr[e \in E(L_i, R_i)] &\geq \frac{1}{4} \\ Pr[e \in F | e \in E(L_i, R_i)] &\geq \left(1 - \frac{8}{\sqrt{d_{i+1}}}\right) \cdot \frac{1}{d_{i+1}} \cdot \left(1 - \frac{2}{d_{i+1}}\right) \end{aligned}$$

Now $Pr[e \in E(R_i, L_i)] = \frac{1}{4}$. Conditioned on the event that $e \in E(R_i, L_i)$, let r be the multiplicity of the anti-parallel arc to e in $E(L_i, R_i)$ after the addition of dummy edges in Step 3. Then, one of these arcs in selected in N_i with probability $\frac{r}{d_{i+1}}$.

$$Pr[e \in M_i | e \in E(R_i, L_i)] \geq \left(1 - \frac{8}{\sqrt{d_{i+1}}}\right) \cdot \frac{r}{d_{i+1}} \cdot \frac{2}{d_{i+1}}$$

In the last case, e is included in G_{i+1} if one of the anti-parallel edges is not chosen in N_i but we chose $M_i = N_i$

$$Pr[e \in G_{i+1} | e \in E(R_i, L_i)] \geq \left(1 - \frac{8}{\sqrt{d_{i+1}}}\right) \cdot \left(1 - \frac{r}{d_{i+1}}\right) \cdot \left(1 - \frac{2}{d_{i+1}}\right)$$

By induction, we have

$$Pr[e \in F | e \in G_{i+1}] \geq \frac{1}{d_{i+1} + f(i+1)\sqrt{d_{i+1}}}$$

where we let $f(j) = 2^{j/2+1} + 30(\sqrt{2}+1)$ for any j for ease of notation.
Combining all the above inequalities, we obtain that

$$
\begin{aligned}
&Pr[e \in F | e \in G_i] \geq \\
&\frac{1}{4} \cdot \left(1 - \frac{8}{\sqrt{d_{i+1}}}\right) \cdot \frac{1}{d_{i+1}} \cdot \left(1 - \frac{2}{d_{i+1}}\right) \\
&+ \frac{1}{4} \cdot \left(1 - \frac{8}{\sqrt{d_{i+1}}}\right) \cdot \frac{r}{d_{i+1}} \cdot \frac{2}{d_{i+1}} \\
&+ \frac{1}{4} \cdot \left(1 - \frac{8}{\sqrt{d_{i+1}}}\right) \cdot \left(1 - \frac{r}{d_{i+1}}\right) \cdot \left(1 - \frac{2}{d_{i+1}}\right) \cdot \frac{1}{d_{i+1} + f(i+1)\sqrt{d_{i+1}}}
\end{aligned}
$$

We first notice that the coefficient at r is always positive and hence the expression is minimized when $r = 0$. Simplifying we get

$$
\begin{aligned}
Pr[e \in F | e \in G_i] &\geq \frac{1}{4} \cdot \left(1 - \frac{8}{\sqrt{d_{i+1}}}\right) \left(1 - \frac{2}{d_{i+1}}\right) \left(\frac{1}{d_{i+1}} + \frac{1}{d_{i+1} + f(i+1)\sqrt{d_{i+1}}}\right) \\
&\geq \frac{1}{4} \cdot \left(1 - \frac{10}{\sqrt{d_{i+1}}}\right) \left(\frac{1}{d_{i+1}} + \frac{1}{d_{i+1} + f(i+1)\sqrt{d_{i+1}}}\right) \\
&\geq \frac{1}{4} \left(\frac{1}{d_{i+1} + 30\sqrt{d_{i+1}}} + \frac{1}{d_{i+1} + f(i+1)\sqrt{d_{i+1}}}\right)
\end{aligned}
$$

where the last inequality is true for large enough d. Using the fact that $d_{i+1} = \frac{d_i}{2}$ and simplifying the above expression further, a simple check shows that

$$
\begin{aligned}
Pr[e \in F | e \in G_i] &\geq \frac{1}{2d_i + 60\sqrt{2}\sqrt{d_i}} + \frac{1}{2d_i + 2\sqrt{2}f(i+1)\sqrt{d_i}} &(7)\\
&\geq \frac{2}{2d_i + \left(30\sqrt{2} + \sqrt{2}f(i+1)\right)\sqrt{d_i}} &(8)\\
&= \frac{1}{d_i + \left(15\sqrt{2} + \frac{f(i+1)}{\sqrt{2}}\right)\sqrt{d_i}} &(9)\\
&\geq \frac{1}{d_i + f(i)\sqrt{d_i}} &(10)\\
&= \frac{1}{d_i + \left(2^{\frac{i}{2}+1} + 30(\sqrt{2}+1)\right)\sqrt{d_i}} &(11)
\end{aligned}
$$

where inequality (8) follows from the inequality $\frac{1}{x} + \frac{1}{y} \geq \frac{2}{(x+y)/2}$ whenever $x, y > 0$. Inequality (10) follows from the inequality that $f(i) \geq 15\sqrt{2} + \frac{f(i+1)}{\sqrt{2}}$ which can be simply verified since $f(j) = 2^{\frac{j}{2}+1} + 30(\sqrt{2}+1)$ for any j. This completes the proof of lemma by induction.

Using the fact $d_1 \geq \frac{d}{2}$, we obtain that

$$Pr[e \in F] = Pr[e \in F | e \in G_1] \geq \frac{1}{d_1 + \left(2^{\frac{3}{2}} + 30(\sqrt{2}+1)\right)\sqrt{d_1}} \geq \frac{2}{d + 120\sqrt{d}} \quad (12)$$

This completes the proof of Theorem 7.

3.1 Extensions to Nearly Regular Graphs

Here we prove Theorem 5.

Proof. Let G be a connected graph of maximum degree Δ, average degree d, and minimum degree δ. Simply replacing d by Δ in the proof of Theorem 3 establishes that the fractional linear arboricity of G is $\frac{\Delta}{2} + O(\sqrt{\Delta})$. As G has $dn/2$ edges, this implies that it has at least one linear forest F with $n(\frac{d}{\Delta} - O(\frac{1}{\sqrt{\Delta}}))$ edges. Moreover, such a linear forest can be found in polynomial time by sampling linear forests from the distribution generated by the algorithm appearing in the proof of Theorem 3. Given such a linear forest, Corollary 1 finds a tour of length $2n - |F| + O(\frac{n}{\delta}) = \left(1 + \frac{\Delta - d}{\Delta} + O\left(\sqrt{\frac{1}{\Delta}}\right) + O(\frac{1}{\delta})\right) n$, as specified in Theorem 5.

The term $\frac{\Delta - d}{\Delta}$ in Theorem 5 is best possible, as shown by the following example. Let G be a bipartite graph in which the larger side has $\frac{\Delta n}{\Delta + \delta}$ vertices of degree δ, whereas the smaller side has $\frac{\delta n}{\Delta + \delta}$ vertices of degree Δ. As a tour must visit every vertex in the large side, the length of the shortest tour is at least $\frac{2\Delta n}{\Delta + \delta}$. The average degree is $d = \frac{2\Delta\delta}{\Delta + \delta}$, and a simple manipulation show that expressing the minimum tour length as a function of d gives $\left(1 + \frac{\Delta - d}{\Delta}\right) n$, as desired.

The term $O\left(\sqrt{\frac{1}{\Delta}}\right)$ in Theorem 5 is carried over from Theorem 2, and possibly can be replaced by $O(\frac{1}{\Delta})$. See discussion in Section 4.

An interesting question is whether the term $O(\frac{1}{\delta})$ can be replaced by $O(\frac{1}{d})$. If so, the bound in Theorem 5 would become independent of δ. Our proof of Theorem 5 uses Corollary 1, and there the term $O(\frac{1}{\delta})$ cannot be replaced by $O(\frac{1}{d})$. Consider for example a path of length (roughly) $n/2$ in which one endpoint is connected to a triangle and the other to a clique of size $n/2$. This graph has a Hamiltonian path (and hence a linear forest of size $n-1$), but the shortest tour is of length roughly $3n/2$, despite the fact that its average degree is very high, roughly $n/4$. However, the above graph does not show that the term $O(\frac{1}{\delta})$ cannot be replaced by $O(\frac{1}{d})$ in Theorem 5, because for this graph $\frac{\Delta - d}{\Delta} \simeq \frac{1}{2}$.

4 Some Conjectures

Linear arboricity of a graph G is a covering of all its edges by linear forests. The linear arboricity conjecture of [1] states that every d-regular graph has a linear arboricity with $\lceil \frac{d+1}{2} \rceil$ linear forests. If true, then one of these forests must be of size at least $(1 - \frac{2}{d+2})n$, and Theorem 4 would then imply that every d-regular graph has a tour of length $(1 + O(\frac{1}{d}))n$. The linear arboricity conjecture has been proved for small values of d, and is known to be true up to low order terms for large values of d (see [3] or [4]). The known upper bounds on the linear arboricity number translate to a $(1 - O\left(\left(\frac{\log d}{d}\right)^{1/3}\right)n$ lower bound on the sizes of linear forests, which is weaker than the bound that we prove in Theorem 3.

The path cover number of a graph G is the minimum number of vertex disjoint paths required to cover the vertices of G. Magnant and Martin [12] conjecture that the path cover number of d-regular graphs is at most $\frac{n}{d+1}$ (and even smaller if the graph is required to be connected). They prove the conjecture for all $d \leq 5$. Observe that every path cover is a linear forest, and that the size of the forest plus the respective cover number is exactly n. Hence the path cover number conjecture, if true, could be combined with our Corollary 1 to show that every d-regular graph has a tour of length $(1 + O(\frac{1}{d}))n$.

A *minimum Hamiltonian completion* of a graph is the minimum size-set of edges that, when added to the graph, makes it Hamiltonian [10]. The size of such a set is exactly one more than the size of a minimum path cover of the graph.

An upper bound of $(1 + O(\frac{1}{d}))n$ on the shortest tour length is the best that one can hope for, and likewise, a lower bound of $(1 - \Omega(\frac{1}{d}))n$ on the largest linear forest would be best possible. This can be demonstrated by taking a d-regular tree of depth $\ell \simeq \log_d n$ (the root node has d children whereas internal nodes have $d - 1$ children), and converting it to a d-regular graph as follows (assume for simplicity that d is odd). Add a single child to each leaf, connect this child to every sibling of its parent leaf (by now original leaves have degree d), and add a matching on the set of newly added children in which two such vertices can be matched if they are children of sibling leaves (so now all vertices have degree d). In this d-regular graph, a path can contain at most two vertices from the penultimate level of the tree (the parents of the leaves). It follows that a path cover contains at least $\Omega(n/d)$ paths, implying the desired lower bound on the length of a tour and upper bound on size of the linear forest.

Acknowledgements

Part of this work was done while the first two authors were visiting Microsoft Research in Redmond, Washington. We thank Noga Alon and Jeff Kahn for directing us to relevant literature.

References

1. Jin Akiyama, Geoffrey Exoo and Frank Harary. Covering and Packing in Graphs IV: linear Arboricity. Networks, Vol. 11 (1981) 69–72.
2. Noga Alon. Problems and results in extremal combinatorics, Part I. Discrete Math. 273 (2003), 31–53.
3. Noga Alon and Joel Spencer. The Probabilistic Method.
4. N. Alon, V. J. Teague and N. C. Wormald. Linear arboricity and linear k-arboricity of regular graphs. Graphs and Combinatorics 17 (2001), 11–16.

5. Sylvia Boyd, Rene Sitters, Suzanne van der Ster, and Leen Stougie. Tsp on cubic and subcubic graphs. *Proceedings of IPCO*, 2011.
6. N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.
7. José R. Correa, Omar Larré, and José A. Soto. TSP tours in cubic graphs: beyond 4/3. Proceedings of the 20th Annual European conference on Algorithms (ESA'12), 790–801, 2012
8. G. P. Egorychev. The solution of van der Waerden's problem for permanents, Advances in Mathematics 42.3 (1981): 299-305.
9. D. I. Falikman. Proof of the van der Waerden conjecture regarding the permanent of a doubly stochastic matrix, Mathematical Notes 29.6 (1981): 475-479.
10. D. S. Franzblau, and A. Raychaudhuri. Optimal Hamiltonian completions and path covers for trees, and a reduction to maximum flow. ANZIAM Journal 44 (02): 193204, 2002.
11. Mark Jerrum, Alistair Sinclair and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. J. ACM 51(4): 671–697 (2004).
12. Colton Magnant and Daniel M. Martin. A note on the path cover number of regular graphs, Australasian Journal of Combinatorics. Volume 43 (2009), Pages 211-217.
13. Tobias Mömke and Ola Svensson. Approximating graphic tsp by matchings (best paper in FOCS 2011). *CoRR*, abs/1104.3090, 2011.
14. J. W. Moon. On the diameter of a graph, The Michigan Mathematical Journal 12.3 (1965): 349-351.
15. Robin A. Moser and Gabor Tardos. A constructive proof of the general lovsz local lemma. J. ACM 57(2) (2010)
16. Amin Saberi Shayan Oveis Gharan and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In *FOCS*, 2011.
17. András Sebö and Jens Vygen. Shorter Tours by Nicer Ears. *CoRR*, abs/1201.1870, 2012.
18. Nisheeth K. Vishnoi. A Permanent Approach to the Traveling Salesman Problem. FOCS 2012: 76–80.

A Weaker Deterministic Algorithm

In this section, we present an alternate deterministic algorithm for finding a large linear forest with slightly weaker guarantees.

Theorem 8. *Let G be an n-vertex d-regular graph. Then G has a linear forest of size $(1 - O(\sqrt{\frac{\log d}{d}}))n$. Moreover, such a linear forest can be found in polynomial time.*

Proof. Orient all edges of G such that every vertex has indegree and outdegree equal to $d/2$ (when d is even), or either $(d-1)/2$ or $(d+1)/2$ (when d is odd). This can be achieved as described earlier by orienting the edges according to a directed Eulerian tour through G (when d is even), or through G plus an arbitrary perfect matching on non-edges of G (when d is odd).

We shall now describe an iterative algorithm for constructing a linear forest. Each iteration i will involve a corresponding error term ϵ_i. We shall later explain how to control these error terms. We shall assume for simplicity that n is a power of 2. This assumption has negligible effect on our bounds.

Iteration i for $i = 0, 1, 2, \ldots$ starts with $2^{-i}n$ vertices, where every vertex has indegree and outdegree in the range $2^{-i-1}d(1 \pm \epsilon_i)$. Hence $\epsilon_0 \leq 1/d$. The algorithm partitions the vertices into two equal size sets A_i and B_i subject to the following conditions.

- For every $v \in A_i$ its outdegree into B_i is in the range $2^{-i-2}d(1 \pm \epsilon_{i+1})$ and its indegree from B_i is in the range $2^{-i-2}d(1 \pm \epsilon_{i+1})$. Likewise, for every $v \in B_i$ its outdegree into A_i is in the range $2^{-i-2}d(1 \pm \epsilon_{i+1})$ and its indegree from A_i is in the range $2^{-i-2}d(1 \pm \epsilon_{i+1})$.

The algorithm now considers only the edges directed from A_i to B_i, denoted by $E(A_i, B_i)$, and finds a maximum matching M_i with respect to these edges.

Proposition 2. *The maximum matching M_i over $E(A_i, B_i)$ satisfies $|M_i| \geq (1 - 2\epsilon_{i+1})|A_i|$.*

Proof. Give every edge of $E(A_i, B_i)$ a weight of $\frac{2^{i+2}}{(1+\epsilon_{i+1})d}$. The sum of weights incident to a vertex is at most 1, and hence these weights form a feasible point in the bipartite matching polytope. It follows that there is a bipartite matching of at least the same total weight, which is at least $|A_i|\frac{1-\epsilon_{i+1}}{1+\epsilon_{i+1}} \geq (1 - 2\epsilon_{i+1})|A_i|$.

Complete the matching M_i to a perfect matching between A_i and B_i in an arbitrary way (adding auxiliary edges between A_i and B_i if needed). For every pair of matched vertices $u \in A_i$ and $v \in B_i$, merge them to one new vertex, whose outgoing edges are those edge going out of v, and its incoming edges are those incoming to u. If the edge (v, u) exists, remove it. This completes the description of iteration i. Observe that the number of vertices, their indegrees and outdegrees at the end of iteration i are exactly as required to be in the beginning of iteration $i + 1$.

The algorithm ends when no edges remain in the graph. When it ends, it outputs the union of all M_i.

Proposition 3. *The output of the algorithm is a linear forest.*

Proof. Add to the output of the algorithm also all the auxiliary edges that were used in order to extend the M_i matchings into perfect matchings. Considering the graph induced on the edges of all M_i and the auxiliary edges, it can be verified by induction that every vertex in iteration i corresponds to a directed path with exactly 2^i vertices, where the operation performed in iteration i is matching such paths in pairs, and concatenating the two members of a pair. Hence with the auxiliary edges the final output of the algorithm is a Hamiltonian path. Removing the auxiliary edges leaves a vertex disjoint set of paths (some of which might be isolated vertices), which by definition is a linear forest.

The size of the linear forest output by the algorithm is at least

$$\sum_{i \geq 0} M_i \geq \sum_{i \geq 1} (1 - 2\epsilon_i) n 2^{-i} = n - 1 - 2n \sum_{i \geq 1} \frac{\epsilon_i}{2^i}$$

where the middle inequality follows from Proposition 2 and straightforward change of indices.

Lemma 3. *For some universal constant c independent of n and d, there is a polynomial time version of the algorithm above that ensures for every $i \geq 1$ that $\epsilon_i \leq c 2^{i/2}\sqrt{\frac{\log d}{d}}$.*

Proof. We prove the lemma by induction on i. The base of the induction is served by ϵ_0 which was already noted to be at most $1/d$. We now proceed with the inductive step. Observe that ϵ_i measures the deviation of degrees (outdegree or indegree) of vertices of A_i (and B_i) from a postulated average value of $d2^{-i-2}$. We shall use the Lovasz local lemma, or rather, its algorithmic version [15], since we want our results to be algorithmic. At the beginning of iteration i, arrange the vertices in pairs arbitrarily. Now from each pair, place one vertex in A_i and the other in B_i, randomly and independently across pairs. In expectation, the out-degree and in-degree of each vertex is halved. A bad event is one in which the (out/in)-degree of vertex v deviates from its expectation by more than $c'\sqrt{\frac{\log D}{D}}$ times its expectation where we let $D = 2^{-i}d$. Observe that vertices have degree at least $\frac{D}{4}$ and at most $D = 2^{-i}d$. Thus such an event depends on at most $O(D^2)$ other events (only on events involving either a neighbor of v or a neighbor of a neighbor). If the constant c is sufficiently large, standard Chernoff bounds imply that the probability that a bad event happens is $O(D^{-3})$ for appropriately chosen constant c'. Hence the local lemma implies that there is a choice making none of the bad events happen, and the algorithmic version of the local lemma provides an algorithm for finding such a partition into A_i and B_i.

Now we bound the error obtained due to this partition. There are three sources for the error ϵ_i.

1. An error inherited from ϵ_{i-1}. As degrees are halved between iterations, this error is halved as well. However, as the error is relative to the new degrees which are halved, the net effect is that ϵ_{i-1} is inherited as is.
2. A possible loss of 1 in the degree due to a removal of an edge (v, u) between vertices matched in M_{i-1}. This effect is negligible compared to the other terms contributing to ϵ_{i+1}.
3. An error since we did not partition the degree exactly by half and achieved an additional multiplicative error of $c'\sqrt{\frac{\log D}{D}}$.

Thus the relevant degree of each vertex after this iteration is in the range $2^{-i-2}d(1 \pm \epsilon_i)(1 \pm c'\sqrt{\frac{\log D}{D}}) \pm 1$. Ignoring the lower order terms, which can be simply absorbed in any of the inequalities that follow, we obtain

$$\epsilon_{i+1} \leq \epsilon_i + c'\sqrt{\frac{\log D}{D}} \tag{13}$$

$$\leq c2^{i/2}\sqrt{\frac{\log d}{d}} + c'2^{i/2}\sqrt{\frac{\log d - i}{d}} \tag{14}$$

$$\leq c2^{i/2}\sqrt{\frac{\log d}{d}} + \frac{c}{4}2^{i/2}\sqrt{\frac{\log d}{d}} \tag{15}$$

$$\leq c2^{(i+1)/2}\sqrt{\frac{\log d}{d}} \tag{16}$$

where we obtain the first inequality by induction and the definition of $D = 2^{-i}d$, second inequality follows from c being sufficiently large. Thus the error term remains below the upper bound stated in the lemma.

Thus the number of edges in linear forest F is at least

$$|F| \geq n - 1 - 2n \sum_{i \geq 1} \frac{\epsilon_i}{2^i} \tag{17}$$

$$\geq n - 1 - 2cn\sqrt{\frac{\log d}{d}} \sum_{i \geq 1} \frac{1}{2^{i/2}} \tag{18}$$

$$\geq n - 1 - 8cn\sqrt{\frac{\log d}{d}} \tag{19}$$

and thus proving Theorem 8.