Average-Case Complexity of Shortest-Paths Problems in the Vertex-Potential Model*

Colin Cooper,[1] Alan Frieze,[2,†] Kurt Mehlhorn,[3,‡] Volker Priebe[3,‡]

[1] *School of Mathematical Sciences, University of North London, London N7 8DB, UK*

[2] *Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh PA 15213*

[3] *Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany*

ABSTRACT: We study the average-case complexity of shortest-paths problems in the vertex-potential model. The vertex-potential model is a family of probability distributions on complete directed graphs with arbitrary real edge lengths, but without negative cycles. We show that on a graph with n vertices and with respect to this model, the single-source shortest-paths problem can be solved in $O(n^2)$ expected time, and the all-pairs shortest-paths problem can be solved in $O(n^2 \log n)$ expected time. © 2000 John Wiley & Sons, Inc.   Random Struct. Alg., 16, 33–46, 2000

1. INTRODUCTION

A large variety of combinatorial-optimization problems can be modeled as *shortest-paths problems*. Given a (directed) graph in which edges are assigned real edge lengths, these problems ask for *distances* between pairs of vertices. The distance of vertex j from vertex i is defined as the infimum of the lengths of all directed paths from i to j, where the length of a path is the sum of the lengths of its edges. [We need to take the infimum, since in the presence of a *negative cycle*, that is, a (directed) cycle of negative length, a finite minimum does not always exist.] We concentrate on two types of shortest-paths problems. In the *single-source shortest-paths problem*, we are interested in the distances of all vertices from a given source vertex; in the *all-pairs shortest-paths problem*, we want to compute the distance between each pair of vertices.

The *worst-case complexity* of known algorithms for the single-source shortest-paths problem on a directed graph with vertex set $[n] = \{1, \ldots, n\}$ and m edges depends heavily on whether or not edge lengths are allowed to be negative. In fact, if all edge lengths are nonnegative, then Dijkstra's algorithm solves the single-source shortest-paths problem in near-linear time $O(m + n \log n)$ [8, 11]. In the general case, the Bellman–Ford algorithm [2, 10] solves the single-source shortest-paths problem in time $O(\nu m)$, where ν is the maximum number of edges on a shortest path. This is $O(nm)$ in the worst case. The solution of *one* single-source shortest-paths problem allows us to transform a problem with arbitrary real edge lengths into an equivalent problem with nonnegative edge lengths [9, 17]. This gives a running time of $O(nm + n^2 \log n)$ for the all-pairs shortest-paths problem in the general case. The algorithms of McGeoch [21] and Karger, Koller, and Phillips [18] solve the all-pairs shortest-paths problem with nonnegative edge lengths in time $O(n|H| + n^2 \log n)$, where H is the set of edges that are a shortest path between their endpoints. Somewhat better running times are known if the edge lengths are assumed to be integers from some fixed range; see [1, 13, 5].

Worst-case analysis, however, sometimes fails to bring out the advantages of algorithms that perform well in practice; average-case analysis has turned out to be more appropriate for these purposes. In *average-case analysis*, we study the expected running time of shortest-paths algorithms, where instances of shortest-paths problems are generated according to a probability distribution on the set of complete directed graphs with edge lengths. Two kinds of probability distributions have been considered in the literature. In the *uniform model*, the edge lengths are independent, identically distributed random variables. The *endpoint-independent model* is more general. A random instance generated according to the endpoint-independent model has the property that if the edges leaving a vertex are sorted according to their lengths, then the associated endpoints occur in random order. This even includes the case that the edge lengths themselves are arbitrarily fixed, and only the assignment of the edge lengths to the edges leaving a vertex is random.

The average running time of shortest-paths algorithms has mainly been studied for the case of nonnegative edge lengths. In the endpoint-independent model, the following results on the average-case complexity of the single-source shortest-paths problem (on instances with n vertices) are known. Noshita [24] analyzed the average-case complexity of Dijkstra's algorithm; the time bound, however, does not improve over the worst-case complexity of the algorithm. Spira [26] dealt first with

the average-case complexity of the all-pairs shortest-paths problem. He proved an expected time bound of $O(n^2(\log n)^2)$, which was later improved by Bloniarz [3] and Frieze and Grimmett [12]. Moffat and Takaoka [23] described an algorithm with expected running time $O(n^2 \log n)$. Mehlhorn and Priebe [22] showed that the algorithm of Moffat and Takaoka is reliable, that is, it runs in time $O(n^2 \log n)$ with high probability and not just in expectation. In the uniform model, Frieze and Grimmett [12] derived precise results on the distribution of edge lengths and distances. It is a consequence of their results that the expected running time of the algorithms of McGeoch and Karger et al. is $O(n^2 \log n)$ in the uniform model.

Concerning the average-case complexity of the shortest-paths problem in the case of arbitrary real edge lengths, it is already difficult to define a probability distribution that allows negative edge lengths, but does not trivialize the problem. For an instance of the shortest-paths problem on a directed graph D, let $D_{<0}$ and $D_{\geq 0}$ be the subgraphs formed by the edges of negative and nonnegative length, respectively. If $D_{<0}$ contains a cycle and $D_{\geq 0}$ consists of a single strongly connected component, then all distances are $-\infty$. Topological sorting allows us to decide in linear time whether $D_{<0}$ contains a cycle, and depth-first search allows us to decide in linear time whether $D_{\geq 0}$ consists of a single strongly connected component. We conclude that shortest-paths problems become trivial if the probability distribution is such that, with high probability, $D_{<0}$ contains a cycle and $D_{\geq 0}$ consists of a single strongly connected component. This will, for example, be the case if edge lengths have a (constant) nonzero probability of being negative and instances are generated according to the uniform model.

To the best of our knowledge, a probability distribution for graphs with arbitrary real edge lengths (to generate instances of shortest-paths problems) was proposed for the first time by Kolliopoulos and Stein [19]. They studied the endpoint-independent model and gave an algorithm for the single-source shortest-paths problem with expected running time $O(n^2 \log n)$. In general, it is not clear how likely a negative cycle is in the endpoint-independent model. (Note, however, that if each vertex has at least one outgoing edge of negative length, then the directed graph must contain a negative cycle.)

We study the average-case complexity of shortest-paths problems in the *vertex-potential model*. This model was used previously by Cherkassky, Goldberg, and Radzik [4] in an experimental evaluation of shortest-paths algorithms. In this model, there is a *potential* π_i for each vertex $i \in [n]$ and a random variable $r_{i,j}$ for each edge (i, j), $i, j \in [n]$. (We will set $r_{i,i} \equiv 0$ for $i \in [n]$.) The edge lengths are defined by

$$c_{i,j} = r_{i,j} - \pi_i + \pi_j \qquad \text{for all edges } (i, j).$$

Of course, only the $c_{i,j}$s are revealed to our algorithms and the $r_{i,j}$s and π_is are hidden parameters of the model. The variables $r_{i,j}$, $i, j \in [n]$, $i \neq j$, are assumed to be independent, identically distributed random variables with values in $[0, 1]$. (Some additional assumptions on the common distribution function F of the $r_{i,j}$s are needed to allow the application of the results of Frieze and Grimmett [12]. This will be made more precise in Section 3.) The assumption $r_{i,j} \geq 0$, $i, j \in [n]$, guarantees that instances generated according to the vertex-potential model contain no negative cycle; see Propositions 1 and 2. The vertex potentials π_i, $i \in [n]$, may be arbitrarily chosen. (This is a considerable generalization of the model that we

studied in [6], where the π_is were assumed to be random variables with values in $[-1, 1]$.)

We show that the single-source shortest-paths problem can be solved in $O(n^2)$ expected time and that the all-pairs shortest-paths problem can be solved in $O(n^2 \log n)$ expected time. In both cases our algorithms are reliable, that is, finish their computations within the respective time bounds with high probability. Our algorithms are tailored to random inputs generated according to the vertex-potential model. They exploit two structural properties that are exhibited by the input graphs with high probability. First, we can reduce the number of edges in the input graphs without changing the shortest-path distances. Second, each shortest path consists of at most $O(\log n)$ edges.

More precisely, our algorithm for the single-source shortest-paths problem computes vertex potentials $\widehat{\pi}_i$, $i \in [n]$, with the property that for any i, $j \in [n]$, the difference $\widehat{\pi}_i - \widehat{\pi}_j$ is close to $\pi_i - \pi_j$. The $\widehat{\pi}_i$s are then used to define reduced edge lengths $\widehat{c}$ by $\widehat{c}_{i,j} := c_{i,j} + \widehat{\pi}_i - \widehat{\pi}_j$. The reduced edge lengths will not be nonnegative. However, since they are good approximations of the $r_{i,j}$s, they allow the extraction of a small number of relevant edges from the complete graph that, with high probability, give the same shortest-path distances. We extract $O(n^2 / \log n)$ relevant edges and run the Bellman–Ford algorithm on this edge set. Due to the second property, the algorithm computes a solution to one single-source shortest-paths problem in expected time $O(n^2)$; see Theorem 11. [For the same reason, the algorithm of Bellman and Ford would solve the single-source shortest-paths problem in expected time $O(n^2 \log n)$ if run on complete input graphs.] The all-pairs shortest-paths problem can be solved in $O(n^2 \log n)$ expected time if we use the solution to one single-source shortest-paths problem to define nonnegative edge lengths and then employ the algorithm of McGeoch or Karger, Koller, and Phillips; see Theorem 13.

The vertex-potential model and the endpoint-independent model are incomparable as the endpoint-independent model cannot exclude negative cycles and the vertex-potential model excludes negative cycles. Moreover, in the vertex-potential model, there is a strong correlation between the endpoint of an edge and the length of the edge. Assume that the distribution function F is such that all $r_{i,j}$s are very close to zero. This implies that for all vertices i, it is probable that the shortest edge leaving i will go to the vertex with minimum potential.

The paper is organized as follows. In Section 2, we recall basic facts about shortest paths and reduced edge lengths. We provide auxiliary results needed for the analysis of our algorithms in Section 3. The algorithms and their analysis are presented in Section 4.

2. SHORTEST PATHS AND REDUCED EDGE LENGTHS

Let D_n be the complete loopless directed graph on the set $[n] = \{1, \ldots, n\}$ of vertices. Assume that edge lengths are given by a function c from the set of edges to the reals; the length of the directed edge (i, j) will be denoted by $c_{i,j}$ for i, $j \in [n]$. (For convenience, let $c_{i,i} \equiv 0$ for $i \in [n]$.) We will write (D_n, c) for D_n with edge lengths c.

For a directed path P in D_n, let $c(P)$ be the length of P with respect to c, that is, $c(P)$ is defined as $\sum_{(i,j)\in P} c_{i,j}$. For any pair i, j of vertices, let $\delta_{i,j}(c)$ be the infimum of the lengths of all paths from i to j. The quantity $\delta_{i,j}(c)$ will be referred to as the *distance* of j from i (with respect to c).

We consider the following two shortest-paths problems. For a given source vertex $s \in [n]$, the single-source shortest-paths problem asks for the distances of all vertices $i \in [n]$ from s. If s is fixed and no confusion is possible, we will denote these distances by $\delta_i(c)$ for $i \in [n]$. In the all-pairs shortest-paths problem, we want to compute the distance between each pair of vertices. The maximum of all these distances is called the *diameter* of D_n with respect to c and will be denoted by $\Delta(c)$.

As mentioned in the Introduction, we want to generate instances of shortest-paths problems without negative cycles. The following proposition characterizes edge lengths c for which (D_n, c) does not contain any negative cycle.

Proposition 1. *The absence of negative cycles in (D_n, c) is equivalent to the existence of vertex potentials $\pi_i \in \mathbb{R}$, $i \in [n]$, so that the reduced edge lengths r (of c with respect to the π_is) are nonnegative, that is,*

$$r_{i,j} := c_{i,j} + \pi_i - \pi_j \geq 0 \quad \text{for all edges } (i, j).$$

Proposition 1 relies on the following *optimality conditions* for a solution to the single-source shortest-paths problem (with source s).

For every vertex $i \in [n]$, let $d_i(c)$ denote the length (with respect to c) of some directed path from s to i, with $d_s(c) = 0$. The quantities $d_i(c)$ are equal to the distances $\delta_i(c)$ if and only if they satisfy

$$d_i(c) + c_{i,j} \geq d_j(c) \qquad \text{for all edges } (i, j). \tag{2.1}$$

It is well known that shortest-paths problems are invariant under reduction of edge lengths with respect to vertex potentials. We summarize this knowledge in the following proposition.

Proposition 2. *Suppose that we associate a vertex potential $\widehat{\pi}_i \in \mathbb{R}$ with each vertex $i \in [n]$ and that we define reduced edge lengths $\widehat{c}$ (of c with respect to the $\widehat{\pi}_i$s) by $\widehat{c}_{i,j} := c_{i,j} + \widehat{\pi}_i - \widehat{\pi}_j$ for any edge (i, j). Then, for any directed path P from vertex k to vertex l,*

$$\widehat{c}(P) = \sum_{(i,j)\in P} (c_{i,j} + \widehat{\pi}_i - \widehat{\pi}_j) = c(P) + \widehat{\pi}_k - \widehat{\pi}_l. \tag{2.2}$$

Therefore, distances with respect to c and $\widehat{c}$ relate to each other through $\delta_{k,l}(\widehat{c}) = \delta_{k,l}(c) + \widehat{\pi}_k - \widehat{\pi}_l$ for all k, $l \in [n]$. It also follows from (2.2) that, for any directed cycle C, $\sum_{(i,j)\in C} \widehat{c}_{i,j} = \sum_{(i,j)\in C} c_{i,j}$; in particular, $(D_n, \widehat{c})$ contains a negative cycle if and only if there is a negative cycle in (D_n, c).

3. PROPERTIES OF THE VERTEX-POTENTIAL MODEL

In the vertex-potential model that we consider in this paper, edge lengths $c_{i,j}$ are generated according to

$$c_{i,j} = r_{i,j} - \pi_i + \pi_j \qquad \text{for all edges } (i, j),$$

with random variables $r_{i,j} \geq 0$ and (not necessarily random) vertex potentials π_i, i, $j \in [n]$. (We will set $r_{i,i} \equiv 0$ for $i \in [n]$.) Our precise assumptions for these variables are as follows:

  (A1) The variables $r_{i,j}$, i, $j \in [n]$, $i \neq j$, are independent, identically distributed random variables with values in $[0, 1]$ and mean ρ. We will denote their common distribution function by F; it is assumed to be independent of n. We further assume that $F(0) = 0$ and that $F'(0)$ exists and is strictly positive. (This implies $\rho > 0$.)
  (A2) The vertex potentials π_i, $i \in [n]$, are arbitrary real numbers.

The independence assumptions in (A1) are the most important (and restrictive) ones. Assumption (A2) considerably generalizes the assumption on the π_is in [6], where they were assumed to be independent, identically distributed random variables with values in $[-1, 1]$. We will not need to know the value of ρ; if needed, we could get a good estimate for it from the data; see the remark after Lemma 8. This lemma will be the only place where we use that the random variables are bounded, and the assumption of boundedness is there more for convenience than necessity. It could be replaced by bounds on the tails of the distributions. By definition of $F'(0)$ and since $F(0) = 0$,

$$F(\varepsilon) = \Pr(r_{i,j} \leq \varepsilon) = F'(0) \cdot \varepsilon + o(\varepsilon) \qquad \text{as } \varepsilon \downarrow 0, \tag{3.1}$$

that is, the assumption $F'(0) > 0$ implies that the distribution of the $r_{i,j}$s can be approximated by uniform distributions in a neighborhood of 0. This allows reducing the proof of Lemma 4 to the case of the uniform distribution on $[0, 1]$. The proof of the following tail estimate also makes use of this fact.

Lemma 3. *Let G be a fixed distribution function with the properties that $G(0) = 0$ and that $G'(0)$ exists. Then there are constants $R = R_G$ and ε_G such that for any collection $X_1, \ldots, X_m$ of independent random variables, all distributed according to G, and for any $\varepsilon < \varepsilon_G$,*

$$\Pr(X_1 + \cdots + X_m \leq \varepsilon) \leq \left(\frac{eR\varepsilon}{m}\right)^m. \tag{3.2}$$

Proof. By our assumptions on G, $G(\varepsilon) = G'(0) \cdot \varepsilon + o(\varepsilon)$, as $\varepsilon \downarrow 0$, and there exist constants $R = R_G > G'(0)$ and $\varepsilon_G \leq 1/R$ such that, for all $\varepsilon < \varepsilon_G$, $G(\varepsilon) \leq R \cdot \varepsilon$. Let X be a random variable that is distributed according to G and let Y be a random variable that is uniformly distributed on $[0, 1/R]$. For all $\varepsilon < \varepsilon_G$, we have by construction,

$$\Pr(X \leq \varepsilon) = G(\varepsilon) \leq R \cdot \varepsilon = \Pr(Y \leq \varepsilon),$$

which implies that

$$\Pr\big(\min\{X, \varepsilon_G\} \leq \xi\big) \leq \Pr\big(\min\{Y, \varepsilon_G\} \leq \xi\big) \qquad \text{for all } \xi \geq 0.$$

This means that the random variable $\min\{Y, \varepsilon_G\}$ is stochastically dominated by the random variable $\min\{X, \varepsilon_G\}$. Let $Y_1, \ldots, Y_m$ be independent copies of Y. Since stochastic dominance is preserved under taking sums of independent random variables (see, for example, [25, Example 9.2(A)]), for any $\varepsilon < \varepsilon_G$, we get the bound

$$\begin{aligned}
\Pr(X_1 + \cdots + X_m \leq \varepsilon) &= \Pr\big(\min\{X_1, \varepsilon_G\} + \cdots + \min\{X_m, \varepsilon_G\} \leq \varepsilon\big) \\
&\leq \Pr\big(\min\{Y_1, \varepsilon_G\} + \cdots + \min\{Y_m, \varepsilon_G\} \leq \varepsilon\big) \\
&= \Pr(Y_1 + \cdots + Y_m \leq \varepsilon).
\end{aligned} \tag{3.3}$$

It is easily seen that $\Pr(Y_1 + \cdots + Y_m \leq \varepsilon) = (R\varepsilon)^m/m!$ for any $\varepsilon \leq 1/R$, since the probability in question equals the volume of a suitably scaled standard simplex in $\mathbb{R}^m$. We thus get from (3.3) that

$$\Pr(X_1 + \cdots + X_m \leq \varepsilon) \leq \frac{(R\varepsilon)^m}{m!} \leq \left(\frac{eR\varepsilon}{m}\right)^m$$

for any $\varepsilon < \varepsilon_G$, which proves (3.2). ∎

For a problem of size n, we will say that an event occurs *with high probability* if it occurs with probability at least $1 - O(n^{-\gamma})$ for an arbitrary but fixed constant $\gamma \geq 1$. To ensure a probability of failure $O(n^{-\gamma})$, in most of our statements we have to choose sufficiently large constants, depending on the actual value of γ. This is sometimes made explicit by a subscript γ.

3.1. The Nonnegative Case ($\pi \equiv 0$)

If the $r_{i,j}$s are distributed as in (A1) and the π_is are identically 0, then this gives rise to instances (D_n, r) with nonnegative edge lengths, and the analysis of shortest-paths algorithms by Frieze and Grimmett in [12] can be applied. We briefly review how they argue to obtain bounds on the distances in (D_n, r). For every vertex i, they construct a spanning tree T_i rooted at i. With high probability, the length of the path in T_i from the root to any other vertex is $O((\log n)/n)$ [12, (4.6) and (4.14)]. This implies that the diameter $\Delta(r)$ is $O((\log n)/n)$ with high probability.

Furthermore, if (i, j) is the pth shortest edge in the adjacency list of i, where $p > B_\gamma \log n$ (for a sufficiently large constant B_γ), then $r_{i,j} > \Delta(r)$ with high probability [12, Lemma 4.3]. This means that, with high probability, edge (i, j) is *irrelevant*, that is, is not contained in any shortest path in (D_n, r). We can therefore restrict ourselves to examining the sparse graph with only the $O(\log n)$ shortest edges from each adjacency list. We will show in Lemma 10 how to adjust this idea of "sparsifying" the graph if edge lengths are distributed according to the vertex-potential model.

Lemma 4 [12].   *Suppose that edge lengths in (D_n, r) are distributed as specified in (A1). Then, with high probability, $\Delta(r) = O((\log n)/n)$, and the set $A := \{(i, j); r_{i,j} \leq \Delta(r)\}$ of relevant edges is of cardinality $O(n \log n)$.*

(For the special case of uniformly distributed r, similar results were proved by Hassin and Zemel [14].) By a fairly involved argument, Frieze and Grimmett also proved that each of the trees T_i, $i \in [n]$, is of depth $O(\log n)$, that is, that these short (in length, though not necessarily shortest-length) paths in T_i consist of $O(\log n)$ edges with high probability [12, Theorem 5.2]. One might imagine that a considerable fraction of *shortest* paths has more (but shorter) edges, but we now prove that, with high probability, shortest paths also consist of only logarithmically many edges.

Lemma 5. *With high probability, shortest paths in (D_n, r) consist of $O(\log n)$ edges.*

Proof. Recall from Lemma 4 that, for $\gamma \geq 1$ arbitrary but fixed, there is a constant $C = C_\gamma$ such that, with probability at least $1 - O(n^{-\gamma})$, the diameter $\Delta(r)$ in (D_n, r) is at most $C(\log n)/n$. We fix such a constant C, though we will not necessarily assume in the following that $\Delta(r) \leq C(\log n)/n$. Let us call a path P in (D_n, r) *short* if $r(P) = \sum_{(i,j) \in P} r_{i,j} \leq C(\log n)/n =: \varepsilon_s$. The length (with respect to r) of a given (directed) path consisting of a fixed number of k edges is the sum of k independent, identically distributed random variables, and according to (A1), their common distribution function satisfies the assumptions of Lemma 3. For sufficiently large n, Lemma 3 thus implies that a given path with k edges is short with probability at most $(eR\varepsilon_s/k)^k = (eRC \log n)/(nk)^k$ for some constant R that depends only on the distribution function of the edge lengths. This upper bound is at most $(en)^{-k}$ if $k \geq k_0 := K \log n$, for some $K \geq e^2 RC$. Hence, short paths with more than k_0 edges exist in (D_n, r) only with probability at most $\sum_{k \geq k_0}(k+1)!\binom{n}{k+1} \cdot (en)^{-k} \leq n^2 \cdot n^{-K}$. If we also ensure that $K \geq \gamma + 2$, this probability is $O(n^{-\gamma})$. It follows that in (D_n, r), with high probability, $\Delta(r) \leq C(\log n)/n$ *and* all short paths consist of $O(\log n)$ edges. If this is the case, shortest paths in (D_n, r) (with respect to r) are also short (with respect to the definition above), which proves the lemma. ∎

Since shortest paths are invariant under reduction of edge lengths with respect to vertex potentials (see Section 2), the following is an immediate consequence of Lemma 5.

Corollary 6. *Let edge lengths r be distributed as specified in* (A1) *and let edge lengths c be obtained by reducing the edge lengths r with respect to some vertex potentials. Shortest paths in (D_n, c) consist then of $O(\log n)$ edges with high probability.*

The bounds on the quantities with which we are concerned in Lemmas 4 and 5 are essentially tight, as is indicated by the following results. Davis and Prieditis [7] showed that, for exponentially distributed r, the expected length of a shortest path is of order $(\log n)/n + O(1/n)$. This is also true for r uniformly distributed; see [7, 21]. In fact, this result holds if the distribution of r is chosen from a fairly general class of distributions, including both exponential and uniform distributions, as was proved (for undirected graphs) by Janson [16] recently. Janson also studied the number of edges on shortest paths, proving (among other results) that for ν_s the maximum number of edges on shortest paths from a given vertex s, $\nu_s/\log n$ converges in probability to e. A similar result is conjectured in [16] for the maximum number of edges on any shortest path.

3.2. The General Case: Approximating the Vertex-Potential Differences

We will use the following form of the well-known *Chernoff–Hoeffding bound* on the tail of the distribution of a sum of independent random variables; see [15, 20] for a proof.

Lemma 7. *Let $X_1, \ldots, X_m$ be independent, identically distributed random variables with values in $[0, 1]$; let $\xi := E[X_1]$. Then, for any ε, $0 < \varepsilon < 1$,*

$$\Pr\left(\left|\frac{1}{m}(X_1 + \cdots + X_m) - \xi\right| > \varepsilon\xi\right) \le 2 \cdot e^{-\varepsilon^2 m\xi/3}. \tag{3.4}$$

We now show how to compute vertex potentials $\widehat{\pi}_i$, $i \in [n]$, so that, for any $k, l \in [n]$, with high probability, the difference $\widehat{\pi}_k - \widehat{\pi}_l$ is a good approximation of the actual vertex-potential difference $\pi_k - \pi_l$.

Lemma 8. *For any $i \in [n]$, define*

$$\widehat{\pi}_i := -\frac{1}{n} \cdot \sum_{j=1}^{n} c_{i, j}.$$

Then, for any $k, l \in [n]$, the term $|(\widehat{\pi}_k - \widehat{\pi}_l) - (\pi_k - \pi_l)|$ is $O(\sqrt{(\log n)/n})$ with high probability.

Proof. For any $k, j \in [n]$, we have $-c_{k, j} = \pi_k - r_{k, j} - \pi_j$ by definition; thus,

$$-\sum_{j=1}^{n} c_{k, j} = n \cdot \pi_k - \sum_{j=1}^{n} r_{k, j} - \sum_{j=1}^{n} \pi_j,$$

from which we conclude that

$$\widehat{\pi}_k - \pi_k = -\frac{1}{n} \cdot \sum_{j=1}^{n} r_{k, j} - \frac{1}{n} \cdot \sum_{j=1}^{n} \pi_j.$$

Note that the rightmost term, $n^{-1} \cdot \sum_{j=1}^{n} \pi_j$, is independent of k. Hence, for any k, $l \in [n]$, $|(\widehat{\pi}_k - \pi_k) - (\widehat{\pi}_l - \pi_l)| \le (n-1)^{-2} \cdot (\sum_{j=1}^{n} r_{k, j} - \sum_{j=1}^{n} r_{l, j})|$. This implies

$$|(\widehat{\pi}_k - \pi_k) - (\widehat{\pi}_l - \pi_l)| \le 2 \cdot \max_i \left|\frac{1}{n-1} \cdot \sum_{j=1}^{n} r_{i, j} - \rho\right|, \tag{3.5}$$

where $\rho = (n-1)^{-1} \cdot E[\sum_{j=1}^{n} r_{i, j}]$. By assumption (A1), $\sum_{j=1}^{n} r_{i, j}$ is the sum of $n - 1$ independent, identically distributed random variables with values in $[0, 1]$, and hence, the Chernoff–Hoeffding bound (3.4) implies that, with high probability,

$$\left|\frac{1}{n-1} \cdot \sum_{j=1}^{n} r_{i, j} - \rho\right| = O\left(\sqrt{(\rho \cdot \log n)/n}\right) = O\left(\sqrt{(\log n)/n}\right)$$

for any $i \in [n]$. From this we conclude that $2 \cdot \max_i |(n-1)^{-1} \cdot \sum_{j=1}^{n} r_{i,j} - \rho|$ and, by (3.5), $|(\widehat{\pi}_k - \pi_k) - (\widehat{\pi}_l - \pi_l)|$ for any $k, l \in [n]$, are all $O(\sqrt{(\log n)/n})$ with high probability. ∎

Remark 9.   In [6], we defined approximate vertex potentials $\widehat{\pi}_i = \widehat{\rho} - (n-1)^{-1} \cdot \sum_j c_{i,j}$ for $i \in [n]$, where $\widehat{\rho} = \frac{1}{n(n-1)} \cdot \sum_{i,j} c_{i,j} = \frac{1}{n(n-1)} \cdot \sum_{i,j} r_{i,j}$. (The "observed mean" $\widehat{\rho}$ is a good approximation of ρ.) Using the stronger assumptions of [6] on the π_is, one can prove that $|\widehat{\pi}_i - \pi_i| = O(\sqrt{(\log n)/n})$ with high probability. It turns out, however, that the approximation of single vertex potentials is not needed in the analysis of our algorithms; see Section 4.1.

4. THE ALGORITHMS

4.1. Solving the Single-Source Shortest-Paths Problem

We are now ready to explain in detail how we solve an instance (D_n, c) of the single-source shortest-paths problem if edge lengths in (D_n, c) are generated according to the vertex-potential model of Section 3.

Our algorithm proceeds in three phases: a preprocessing phase, a computation phase, and a postprocessing phase, in which the correctness of the solution from the second phase is checked. Let the source vertex for the single-source shortest-paths problem under consideration be denoted by s.

Preprocessing.   The algorithm computes, for every vertex $i \in [n]$, a vertex potential $\widehat{\pi}_i$ as in Lemma 8 and transforms the edge lengths $c_{i,j}$ to

$$\widehat{c}_{i,j} := c_{i,j} + \widehat{\pi}_i - \widehat{\pi}_j = r_{i,j} + (\widehat{\pi}_i - \pi_i) - (\widehat{\pi}_j - \pi_j) \qquad \text{for all edges } (i, j). \quad (4.1)$$

The shortest-paths problems (D_n, c) and $(D_n, \widehat{c})$ are equivalent. However, $(D_n, \widehat{c})$ is more efficiently solvable, since edge lengths $\widehat{c}$ allow a substantial sparsification of the underlying graph. The following lemma gives a bound on the length of edges that are irrelevant with respect to $\widehat{c}$.

Lemma 10.   *If* $\widehat{c}_{i,j} > \Delta(r) + \max_{k,l} |(\widehat{\pi}_k - \pi_k) - (\widehat{\pi}_l - \pi_l)|$, *then edge* (i, j) *is not contained in any shortest path.*

Proof.   If (i, j) is contained in a shortest path, then $\widehat{c}_{i,j} = \delta_{i,j}(\widehat{c}) \le \Delta(\widehat{c})$. It follows from (4.1) and Proposition 2 that for any $k, l \in [n]$, $\delta_{k,l}(\widehat{c}) = \delta_{k,l}(r) + (\widehat{\pi}_k - \pi_k) - (\widehat{\pi}_l - \pi_l)$. This implies $\Delta(\widehat{c}) \le \Delta(r) + \max_{k,l} |(\widehat{\pi}_k - \pi_k) - (\widehat{\pi}_l - \pi_l)|$. ∎

For $\gamma \ge 1$ arbitrary but fixed, we know from Lemma 4 and from Lemma 8 that there exist constants C_γ and M_γ such that

$$\Delta(r) \le C_\gamma (\log n)/n \qquad \text{and} \qquad \max_{k,l} |(\widehat{\pi}_k - \pi_k) - (\widehat{\pi}_l - \pi_l)| \le M_\gamma \sqrt{(\log n)/n}$$

$$(4.2)$$

with probability at least $1 - O(n^{-\gamma})$. For the time being, we will assume that (4.2) holds. For an arbitrary (but fixed) constant $L_\gamma > M_\gamma$ (and sufficiently large n), all *relevant* edges (that is, edges that will possibly be contained in a shortest path) are then contained in

$$\widehat{A} := \left\{ (i, j);\, \widehat{c}_{i, j} \le L_\gamma \sqrt{(\log n)/n} \right\}.$$

(A more careful proof of Lemma 8 reveals that we could set $L_\gamma = 5\sqrt{\gamma}$. This value is not optimal, but nevertheless indicates that, for given γ, some explicit constant L_γ is easily derivable.) The vertex potentials $\widehat{\pi}_i$, $i \in [n]$, the reduced edge lengths $\widehat{c}$, and the set $\widehat{A}$ can be computed in $O(n^2)$ time. Let $\widehat{D}$ denote the graph $([n], \widehat{A})$.

Computation. We now solve a single-source shortest-paths problem with source s on the sparsified graph $(\widehat{D}, \widehat{c})$ by running the Bellman–Ford algorithm [2, 10]. This algorithm maintains tentative distances d_i for every vertex $i \in [n]$. The d_is are initially set to ∞ (except for $d_s = 0$), and d_i always represents the length of some path in $(\widehat{D}, \widehat{c})$ from s to i. The Bellman–Ford algorithm proceeds in passes over the edge set $\widehat{A}$, maintaining the following invariant. After the kth pass, the Bellman–Ford algorithm has correctly computed the distances of all vertices to which there is a shortest path from s consisting of at most k edges. The algorithm actually checks the optimality conditions (2.1) for all edges (in $\widehat{A}$) in each pass, and it will therefore terminate [with all distances in $(\widehat{D}, \widehat{c})$ computed correctly] after the νth pass, where ν is the maximum number of edges on a shortest path in $(\widehat{D}, \widehat{c})$. The running time of the Bellman–Ford algorithm is therefore $O(\nu \cdot |\widehat{A}|)$, which is $O(n^3)$ in the worst case but $o(n^2)$ with high probability, as we now argue.

By (4.1) and (4.2),

$$\widehat{A} \subseteq \left\{ (i, j);\, r_{i, j} \le (L_\gamma + M_\gamma) \sqrt{(\log n)/n} \right\},$$

and it follows from the assumption (3.1) on the distribution of edge lengths that edge (i, j) is an element of the set on the right-hand side with probability $\widehat{p} = \Theta(\sqrt{(\log n)/n})$ for sufficiently large n, independently of all the other edges. The random variable $|\widehat{A}|$, the cardinality of $\widehat{A}$, is therefore stochastically dominated by a random variable that is binomially distributed with parameters $n(n - 1)$ and $\widehat{p}$. We apply the tail estimates of Lemma 7 to deduce that $|\widehat{A}| = O(n(n - 1)\sqrt{(\log n)/n}) = O(n^{3/2}\sqrt{\log n})$ with high probability. By (4.1) and Corollary 6, $\nu = O(\log n)$ with high probability. Hence, with high probability, it takes $O(n^{3/2}(\log n)^{3/2}) = o(n^2)$ time to run the Bellman–Ford algorithm on the sparsified graph $(\widehat{D}, \widehat{c})$.

Postprocessing. The second phase will have failed to compute all distances correctly only if (4.2) does not hold, which happens only with probability $O(n^{-\gamma})$. The optimality conditions (2.1) (checked for all edges) are an $O(n^2)$-time certificate for the correctness of the solution. Since the worst-case running time of the Bellman–Ford algorithm on $(D_n, \widehat{c})$ is $O(n^3)$, we can easily afford to run the Bellman–Ford algorithm on $(D_n, \widehat{c})$ in case of failure, without affecting the bounds on the running time. Finally, it takes $O(n^2)$ time to compute the distances $\delta_i(c) = \delta_i(\widehat{c}) - \widehat{\pi}_s + \widehat{\pi}_i$ for all $i \in [n]$.

The discussion above is summarized in the following theorem.

Theorem 11. *Assume that the edge lengths in (D_n, c) are generated according to the vertex-potential model of Section 3. The single-source shortest-paths problem can then be solved in time $O(n^2)$ with high probability.*

Remark 12. Theorem 11 will still hold if the Bellman–Ford algorithm takes time $O(n^2)$ on $(\widehat{D}, \widehat{c})$ (with high probability), since pre- and postprocessing of the algorithm take time $O(n^2)$ anyway. This means that because of Corollary 6, we could afford to restrict the Bellman–Ford algorithm to the edge set $\{(i, j); \widehat{c}_{i,j} \leq R/\log n\}$, which is a set of cardinality $O(n^2/\log n)$ with high probability for a suitable constant R. In fact, even if we omitted the computation of both vertex potentials and reduced edge lengths and just ran the Bellman–Ford algorithm on the complete input graph, we would solve the single-source shortest-paths problem in time $O(n^2 \log n)$ with high probability because of Corollary 6.

4.2. Solving the All-Pairs Shortest-Paths Problem

Theorem 13. *Assume that the edge lengths in (D_n, c) are generated according to the vertex-potential model of Section 3. The all-pairs shortest-paths problem can then be solved in $O(n^2 \log n)$ with high probability.*

Proof. We first compute distances $\delta_i(c)$, $i \in [n]$, with respect to source vertex 1 by solving a single-source shortest-paths problem as in the proof of Theorem 11 or as indicated in Remark 12. With high probability, this takes time $O(n^2)$ or $O(n^2 \log n)$, respectively. Let $\bar{c}$ be the reduced edge lengths of c with respect to the vertex potentials $\delta_i(c)$, $i \in [n]$, that is, for all edges (i, j),

$$\bar{c}_{i,j} = c_{i,j} + \delta_i(c) - \delta_j(c)$$
$$= r_{i,j} + \big(\delta_i(c) - \pi_i\big) - \big(\delta_j(c) - \pi_j\big). \tag{4.3}$$

It follows from the first equality and the optimality conditions (2.1) that $\bar{c}_{i,j} \geq 0$ for all edges (i, j). The reduced edge lengths $\bar{c}$ can be computed in $O(n^2)$ time, and the same time bound will later allow us to transform distances $\delta_{i,j}(\bar{c})$ into distances $\delta_{i,j}(c)$ for all $i, j \in [n]$.

To compute the $\delta_{i,j}(\bar{c})$s, we run one of the algorithms of Karger, Koller, and Phillips [18] or McGeoch [21] on $(D_n, \bar{c})$, which efficiently solve the all-pairs shortest-paths problem with nonnegative edge lengths. Both algorithms run in time $O(n^2 \log n + n|H|)$, where $H = H(\bar{c})$ is the set of edges that are the shortest path (with respect to $\bar{c}$) between their endpoints. We apply the arguments of Section 2 again. Shortest paths are invariant under reduction of the edge lengths with respect to vertex potentials, and it follows from (4.3) that the edges in H are also the shortest path between their endpoints with respect to edge lengths r. The set H is therefore contained in the set $A := \{(i, j); r_{i,j} \leq \Delta(r)\}$, and it follows by Lemma 4 that $|H| = O(n \log n)$ with high probability. This yields a running time of $O(n^2 \log n)$ with high probability for the algorithms of McGeoch and Karger, Koller, and Phillips. Since $|H| = O(n^2)$ in the worst case, our algorithm has a running time of $O(n^3)$ with probability $O(n^{-\gamma})$, $\gamma \geq 1$, which still gives an expected running time of $O(n^2 \log n)$. ∎

ACKNOWLEDGMENTS

REFERENCES

[1] R.K. Ahuja, K. Mehlhorn, J.B. Orlin, and R.E. Tarjan, Faster algorithms for the shortest path problem, J Assoc Comput Mach 37 (1990), 213–223.

[2] R. Bellman, On a routing problem, Quart Appl Math 16 (1958), 87–90.

[3] P.A. Bloniarz, A shortest-path algorithm with expected time $O(n^2 \log n \log^* n)$, SIAM J Comput 12 (1983), 588–600.

[4] B.V. Cherkassky, A.V. Goldberg, and T. Radzik, Shortest paths algorithms: Theory and experimental evaluation, Math Programming 73 (1996), 129–174.

[5] B.V. Cherkassky, A.V. Goldberg, and C. Silverstein, Buckets, heaps, lists, and monotone priority queues, SIAM J Comput 28 (1999), 1326–1346.

[6] C. Cooper, A. Frieze, K. Mehlhorn, and V. Priebe, "Average-case complexity of shortest-paths problems in the vertex-potential model," Randomization and Approximation Techniques in Computer Science, J. Rolim (Editor), Lecture Notes in Computer Science, vol 1269, Springer-Verlag, Berlin, 1997, pp. 15–26.

[7] R. Davis and A. Prieditis, The expected length of a shortest path, Inform Process Lett 46 (1993), 135–141.

[8] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer Math 1 (1959), 269–271.

[9] J. Edmonds and R.M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, J Assoc Comput Mach 19 (1972), 248–264.

[10] L.R. Ford, Jr., and D.R. Fulkerson, Flows in Networks, Princeton Univ. Press, Princeton, NJ, 1962.

[11] M.L. Fredman and R.E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, J Assoc Comput Mach 34 (1987), 596–615.

[12] A.M. Frieze and G.R. Grimmett, The shortest-path problem for graphs with random arc-lengths, Discrete Appl Math 10 (1985), 57–77.

[13] A.V. Goldberg, Scaling algorithms for the shortest paths problem, SIAM J Comput 24 (1995), 494–504.

[14] R. Hassin and E. Zemel, On shortest paths in graphs with random weights, Math Oper Res 10 (1985), 557–564.

[15] W. Hoeffding, Probability inequalities for sums of bounded random variables, J Amer Statist Assoc 58 (1963), 13–30.

[16] S. Janson, One, two and three times $\log n / n$ for paths in a complete graph with random weights, Combin Probab Comput, to appear.

[17] D.B. Johnson, Efficient algorithms for shortest paths in sparse networks, J Assoc Comput Mach 24 (1977), 1–13.

[18] D.R. Karger, D. Koller, and S.J. Phillips, Finding the hidden path: Time bounds for all-pairs shortest paths, SIAM J Comput 22 (1993), 1199–1217.

[19] S.G. Kolliopoulos and C. Stein, Finding real-valued single-source shortest paths in $o(n^3)$ expected time, J Algorithms 28 (1998), 125–141.

[20] C. McDiarmid, "Concentration," Probabilistic Methods for Algorithmic Discrete Mathematics, M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed (Editors), Algorithms and Combinatorics, vol 16, Springer-Verlag, Berlin, 1998, pp. 195–248.

[21] C.C. McGeoch, All-pairs shortest paths and the essential subgraph, Algorithmica 13 (1995), 426–441.

[22] K. Mehlhorn and V. Priebe, On the all-pairs shortest-path algorithm of Moffat and Takaoka, Random Struct Alg 10 (1997), 205–220.

[23] A. Moffat and T. Takaoka, An all pairs shortest path algorithm with expected time $O(n^2 \log n)$, SIAM J Comput 16 (1987), 1023–1031.

[24] K. Noshita, A theorem on the expected complexity of Dijkstra's shortest path algorithm, J Algorithms, 6 (1985) 400–408.

[25] S.M. Ross, Stochastic Processes, 2nd ed., Wiley, New York, 1996.

[26] P.M. Spira, A new algorithm for finding all shortest paths in a graph of positive arcs in average time $O(n^2 \log^2 n)$, SIAM J Comput 2 (1973), 28–32.