A constant approximation algorithm for the *a priori* traveling salesman problem

David Shmoys[1]* and Kunal Talwar[2]

[1] Cornell University, Ithaca NY 14853
shmoys@cs.cornell.edu
[2] Microsoft Research, Silicon Valley Campus, 1065 L'Avenida, Mountain View, CA 94043
kunal@microsoft.com

Abstract. One of the interesting recent developments in the design and analysis of approximation algorithms has been in the area of algorithms for discrete stochastic optimization problems. In this domain, one is given not just one input, but rather a probability distribution over inputs, and yet the aim is to design an algorithm that has provably good worst-case performance, that is, for *any* probability distribution over inputs, the objective function value of the solution found by the algorithm must be within a specified factor of the optimal value.

The *a priori* traveling salesman problem is a prime example of such a stochastic optimization problem. One starts with the standard traveling salesman problem (in which one wishes to find the shortest tour through a given set of points N), and then considers the possibility that only a subset A of the entire set of points is active. The active set is given probabilistically; that is, there is a probability distribution over the subsets of N, which is given as part of the input. The aim is still to compute a tour through all points in N, but in order to evaluate its cost, we instead compute the expectation of the length of this tour after shortcutting it to include only those points in the active set A (where the expectation is computed with respect to the given probability distribution). The goal is to compute a "master tour" for which this expectation is minimized. This problem was introduced in the doctoral theses of Jaillet and Bertsimas, who gave asymptotic analyses when the distances between points in the input set are also given probabilistically.

In this paper, we restrict attention to the so-called "independent activation" model in which we assume that each point j is active with a given probability p_j, and that these independent random events. For this setting, we give a 8-approximation algorithm, a polynomial-time algorithm that computes a tour whose *a priori* TSP objective function value is guaranteed to be within a factor of 8 of optimal (and a randomized 4-approximation algorithm, which produces a tour of expected cost within a factor of 4 of optimal). This is the first constant approximation algorithm for this model.

---

* Research supported partially by NSF grants CCR-0635121 & DMI-0500263.

1 Introduction

One of the interesting recent developments in the design and analysis of approximation algorithms has been in the area of algorithms for discrete stochastic optimization problems. In this domain, one is not given just one input, but rather a probability distribution over inputs, and yet the aim is to design an algorithm that has provably good worst-case performance, that is, for *any* probability distribution over inputs, the objective function value of the solution found by the algorithm is within a specified factor of optimal value.

The *a priori traveling salesman problem* is a prime example of such a stochastic optimization problem. This is a stochastic generalization of the (standard) traveling salesman problem (TSP). In the traditional deterministic problem, one is given a set N of n points in a metric space; that is, for each pair of points one is given the distance between them, with the assumption that these distances satisfy the triangle inequality (i.e., for each triple of points, i, j, and k, the distance between i and k, is at most the sum of the distances between i and j, and between j and k). The aim is to find the tour (or cyclic permutation) that contains all of these points exactly once, such that the total distance traversed is minimized. In the *a priori* TSP, one is also given a probability distribution Π over subsets $A \subseteq N$; this models the fact that, in advance, one is not certain of the "currently active" points A that need to be included in the tour, and only has probabilistic information about this. However, given only this probabilistic information, one computes a "master tour" τ, whose objective function value is the expected value (with respect to Π) of the length of τ after shortcutting it to include only those active points in A. If the distribution Π makes the full set N active with probability 1, then this is just the usual (deterministic) TSP (and hence we expect results no stronger than is known for that problem).

The *a priori* TSP was first studied in the doctoral theses of Jaillet [1, 2] and Bertsimas [3], and their early work, which mostly focuses on asymptotic analysis when the points themselves are randomly selected (e.g., uniformly in the unit square), is nicely surveyed by Bertsimas, Jaillet, and Odoni [4]. If one is interested in polynomial-time algorithms, then one must be careful in the way that the distribution Π is given as part of the input. It would take an exponential amount of information to list the probability that each set A is active as part of the input; one simple workaround is to insist that only a polynomial (in n) number of sets can be active with a positive probability, and this is called the *polynomial scenario* model. Alternatively, we can present the probability distribution by means of an oracle, a "black box" from which we can draw independent samples according the distribution Π; the restriction that we are interested in polynomial-time algorithms implies that we are limited to a polynomial number of samples in this *black box* model.

In this paper, we restrict attention to the so-called *independent activation* model for Π; for each $j \in N$, consider the random indicator variable, $\mathbb{1}(j \in A)$, which is 1 exactly when $j \in A$, and then we shall assume that these are independent random variables, and that we are given, as part of the input, the probability $p_j = Pr[\mathbb{1}(j \in A) = 1]$, for each $j \in N$. For this setting, we first give a randomized polynomial-time algorithm to compute a tour for the *a priori* TSP whose expected objective function value (with respect to the random choices of the algorithm) is guaranteed to be within a factor of 4 of optimal. We then derandomize the procedure to give a deterministic algorithm that is guaranteed to find a tour of cost within a factor of 8 of optimal. This is the first constant approximation algorithm for this model.

The strongest previous result was a randomized $O(\log n)$-approximation algorithm due to Schalekamp and Shmoys [5]. That result is significantly more general, since not only does it hold in the black box model, but in fact, it does not require *any* knowledge of the distribution Π.

This stochastic notion of the TSP is closely related to the notion of the *universal TSP*. In the universal TSP, there is also a notion of a "master tour" which is shortcut according to the current active set A. However, for this problem, we say that a tour τ for N has a performance guarantee of ρ if, for every subset $A \subseteq N$, the tour τ shortcut to A has length at most ρ times the optimal cost of a tour for the subset A. Clearly, a ρ-approximation algorithm for the universal TSP is a ρ-approximation algorithm for the *a priori* TSP with respect to *any* distribution. The universal TSP was introduced by Bartholdi and Platzman [6] in the context of an application for "meals on wheels", and they gave an $O(\log n)$ performance guarantee for points in the Euclidean plane. In contrast, Hajiaghayi, Kleinberg, and Leighton [7] proved that no tour can achieve a bound better than $c(\log n/\log\log n)^{1/6}$ for some constant $c > 0$, even for points in the Euclidean plane. Thus, it is particularly natural to consider weaker models, such as the *a priori* TSP.

Our algorithm is extremely simple; in essence, we take one sample S from the active set distribution Π, build a minimum spanning tree on that subset, and then add each unselected point as a leaf in this tree by connecting it to its nearest neighbor in S. Then this spanning tree on N is converted to a tour through all points in N, by taking the standard "double tree" walk around the "outside" of the tree. In fact, one must be a bit more careful in handling the case in which the sample S is empty, and this will introduce a number of minor complications. This algorithm is a slight variant on an algorithm that was developed first by Gupta, Kumar, Pál, and Roughgarden [8] in a rather different context, that of the *rent-or-buy problem*, which is a deterministic network design problem in which one can either rent an edge e at a cost of c_e per use, or buy it for unlimited use at a cost of Mc_e (where $M > 0$ is an input to the problem); this algorithm was later also the basis for an algorithm of Gupta, Pál, Ravi and Sinha [9] for the 2-stage (with recourse) stochastic Steiner tree problem. Williamson and van Zuylen [10] have derandomized the rent-or-buy algorithm, and their technique translates almost directly to our setting. Finally, we note that Garg, Gupta, Leonardi, and Sankowski [11, 12], as a corollary of their work on the on-line stochastic Steiner tree, have independently obtained a similar constant approximation algorithm for the *a priori* TSP.

2 The approximation algorithm

2.1 Preliminaries

We start by introducing some notation needed to present our results. For the traveling salesman problem, the input consists of a finite set of points $N = \{1, 2, \ldots, n\}$ as well as the distance $c(i, j)$ between each pair of distinct points $i, j \in N$; we wish to find a tour τ, given by a cyclic permutation of N, such that the length of the tour, $c(\tau) = \sum_{i \in N} c(i, \tau(i))$ is minimized. We shall assume that the distances are symmetric (i.e., $c(j, k) = c(k, j)$ for each pair of points $j, k \in N$) and satisfy the triangle inequality (i.e., $c(j, k) + c(k, \ell) \geq c(j, \ell)$ for each triple of points $j, k, \ell \in N$).

For the *a priori* TSP, only a subset $A \subseteq N$ will be "active". For a tour τ of N, we define τ_A to be the tour "short-cut" to traverse only A; more formally, if we let τ^ℓ be the ℓth power of τ, (i.e., $\tau^1 = \tau$ and for each $\ell > 1$, $\tau^\ell = \tau(\tau^{\ell-1})$), then for each $j \in A$, $\tau_A(j) = \tau^\ell(j)$, where ℓ is the smallest positive integer such that $\tau^\ell(j) \in A$ (unless $A = \{j\}$, in which case $\tau^\ell(j)$ equals j).

For the *a priori* TSP, the input consists of a set of points N, a distance function d, and a probability distribution Π specified over the subsets of N. Since we are working in the independent activation model, we can compute the probability $\Pi(A)$ for each subset $A \subseteq N$; that is,

$$\Pi(A) = \left(\prod_{j \in A} p_j\right)\left(\prod_{j \notin A}(1 - p_j)\right), \tag{1}$$

where p_j is the probability that point j is in the active set A. The goal now is to find a tour τ that minimizes the *expected length* with respect to Π of the tour induced by τ; i.e., to minimize $E_A[c(\tau_A)]$. We shall let τ^* denote an optimal solution, and let OPT denote the corresponding optimal value. It is important to note that τ^*_A denotes the optimal tour shortcut to the subset of nodes A, which is not necessarily the optimal tour for this specific subset.

We will give an 8-approximation algorithm for this problem; that is, we will show how to compute a tour τ such that $E_A[c(\tau_A)] \leq 8 \cdot OPT$. We first give a randomized algorithm that computes a tour τ with the property that $E[E_A[c(\tau)]] \leq 4 \cdot OPT$, where the outer expectation in this expression is with respect to the random coin tosses used by the algorithm, but we will subsequently show how to derandomize this algorithm, while losing an additional factor of 2.

2.2 Special Case: $p_1 = 1$

As a warm-up, we will first consider a more structured case, where $p_1 = 1$.

The randomized algorithm is extremely simple. We first draw a sample S with respect to the underlying distribution Π. In other words, we choose a subset S by independently including every node j in S with probability p_j. We compute a minimum spanning tree on the subset S. Then, we extend this to be a spanning tree T on N, by adding, for each node $j \notin S$, the edge from j to its nearest neighbor in S. (Note that this is well-defined, since we have ensured that $S \neq \emptyset$ by setting $p_1 = 1$.) Let $MST(S)$ denote the cost of the minimum spanning tree (MST) on S and for each $j \neq 1$, let $D_j(S)$ denote the length of this edge between j and its nearest neighbor in $S - \{j\}$. (For notational convenience, let $D_1(S) = 0$.)

From this spanning tree T, it is well known (see, e.g., [13]) that we can construct a tour τ of total length $c(\tau)$ at most twice the total length of edges in the tree T; we simply "walk around the outside of the tree," or more formally, we construct an Eulerian multigraph by taking each edge twice, and then τ is obtained from an Eulerian tour by shortcutting any node that is visited more than once.

To upper bound the objective function value of this tour τ, we will focus on its tree representation; that is, for any subset $A \subseteq N$, the induced tour τ_A can also be obtained by first considering the tree induced from T by the node set A, and then taking the analogous traversal of its doubled Eulerian tour.

In order to analyze the quality of this solution, we next present two obvious lower bounds on the length of an optimal solution.

Fact 1 *For each subset* $A \subseteq N$, $\tau_A^* \geq MST(A)$.

Fact 2 *For each subset* $A \subseteq N$, $\tau_A^* \geq \sum_{j \in A} D_j(A)$.

If we take expectations of both sides of these inequalities, we can thereby obtain lower bounds on the optimal value as well.

Fact 3 $OPT \geq E_A[MST(A)]$.

Fact 4 $OPT \geq E_A[\sum_{j \in A} D_j(A)]$.

Now let us analyze the cost of the (random) tree T generated by our algorithm. Focus on a particular active set A, and its contribution to the expected cost (with respect to Π) of this solution. Recall that T consists of a spanning tree T_S on our sample S, plus additional edges that connect nodes $j \notin S$ to this spanning tree. For a given set A, the induced tree T_A need not include all of the spanning tree T_S; however, for computing an *upper bound* on the cost of T_A, we shall always include all of the edges of T_S in the induced solution. Given this, it is straightforward to see that we can upper bound the cost of T by

$$MST(S) + \sum_{j \neq 1} \mathbb{1}(j \in A)\mathbb{1}(j \notin S) D_j(S), \tag{2}$$

which is bounded above by

$$MST(S) + \sum_{j \neq 1} \mathbb{1}(j \in A) D_j(S). \tag{3}$$

Hence the expected cost (with respect to the random choices of our algorithm) of the tour τ that we compute,

$$E_S[E_A[c(\tau_A)]] \leq 2 \left(E_S[MST(S)] + E_S[E_A[\sum_{j \neq 1} \mathbb{1}(j \in A) D_j(S)]] \right).$$

Since S is a random subset selected according to Π, the first term can be replaced by $E_A[MST(A)]$, which is, by Fact 3, at most OPT. For the second term, we apply linearity of expectation, and the fact that A and S are independent samples drawn according to Π, and hence

$$\begin{aligned} E_S[E_A[\sum_{j \neq 1} \mathbb{1}(j \in A) D_j(S)]] &= \sum_{j \neq 1} E_A[\mathbb{1}(j \in A)] E_S[D_j(S)] \\ &= \sum_{j \neq 1} E_A[\mathbb{1}(j \in A)] E_A[D_j(A)], \end{aligned} \tag{4}$$

where again we have relied on the fact that the subsets S and A are both (independently) drawn according the distribution Π.

For each $j \neq 1$, the random variable $D_j(A)$ denoting the distance from j to its nearest neighbor in $A \setminus \{j\}$ is independent of $\mathbb{1}(j \in A)$. Thus we conclude that

$$\sum_{j \neq 1} E_A[\mathbb{1}(j \in A)] E_A[D_j(A)] = \sum_{j \neq 1} E_A[\mathbb{1}(j \in A) D_j(A)]. \tag{5}$$

On the other hand, recall that by Fact 4,

$$OPT \geq E_A[\sum_j \mathbb{1}(j \in A) D_j(A)] = \sum_j E_A[\mathbb{1}(j \in A) D_j(A)].$$

Thus we can also bound the second term relative to OPT. Combining all of these pieces, we have shown that

$$E_S[E_A[c(\tau_A)]] \leq 2(OPT + OPT) = 4OPT.$$

2.3 General Case

Now we will relax the condition that there must exist one of the points with activation probability equal to 1. One might question where we take advantage of this restriction within the analysis given above. The way in which this gets used is most importantly in the description of the algorithm, in which there always is a set of non-empty points on which to build a spanning tree. Stated another way, for each point j, there is always a point in the sample of the algorithm in $S - \{j\}$. let Π' denote the probability distribution where $\Pi'(A)$ is the probability that A is the active set (as drawn from Π), conditioned on the event that $|A| \geq 2$. For a set A drawn according to Π, let denote the probability that $|A| \geq 2$. The following lemma implies that we can focus on Π' instead of Π.

Lemma 5 *For any tour τ, its* a priori *TSP objective function value with respect to Π is exactly equal to times its objective function with respect to Π'.*

Proof. For each set A with $|A| < 2$, the cost of the short-cut of the tour τ to the subset A, $c(\tau_A)$, is equal to 0. For each subset A with $|A| \geq 2$, the probability that A is active with respect to Π is exactly equal to times the probability that is active with respect to Π'. Now consider the objective function value of τ with respect to Π, which is an expectation computed over the random choice of A. The subsets of size less than two contribute nothing (literally), and the rest contribute in total exactly times what they contribute to the analogous expected value for Π'. Hence, the claim is proved.

This lemma shows that our objective functions with respect to Π and Π' di er only by the same multiplicative factor (for all feasible solutions), and hence we get the following corollary.

Corollary 6 *Any (randomized) ρ-approximation algorithm for the* a priori *TSP with respect Π' is a (randomized) ρ-approximation algorithm with respect to Π.*

Now let us consider the extension of our algorithm and its analysis to Π'. If the algorithms draws a sample S according to Π', we now have the property that for each $j \in N$, there must exist some element in $S - \{j\}$, and hence $D_j(S)$ is well-defined. It is straightforward to see that much of the analysis of the our algorithm is unaffected by this change (though of course, the summation in the bound should now by done over all $j \in N$, not just those nodes $j \neq 1$). The only significant change is that the random variables $\mathbb{1}(j \notin S)$ and $D_j(S)$ are no longer independent so that (5) no longer holds. Note that it would suffice to replace (5) by the inequality

$$\sum_j E_A[\mathbb{1}(j \in A)]E_A[D_j(A)] \leq \sum_j E_A[\mathbb{1}(j \in A)D_j(A)], \tag{6}$$

which, using basic properties of conditional expectation is equivalent to

$$\sum_j E_A[\mathbb{1}(j \in A)]E_A[D_j(A)] \leq \sum_j E_A[\mathbb{1}(j \in A)]E_A[D_j(A)|j \in A]. \tag{7}$$

If we prove that for every $j \in N$, $E_A[D_j(A)] \leq E_A[D_j(A)|j \in A]$, then the remainder of the analysis carries over from the more specialized setting and the desired result follows. The modified algorithm yields a randomized 4-approximation algorithm.

Lemma 7 *For a random set A drawn according to the distribution Π',*

$$E_A[D_j(A)] \leq E_A[D_j(A)|j \in A].$$

Proof. The proof is based on the following two propositions:

Proposition 8 *For a random set A drawn according to the distribution Π',*

$$E_A[D_j(A)|j \notin A] = E_A[D_j(A)|(j \in A) \wedge (|A| \geq 3)].$$

Proof. Let $\mathcal{S}$ be the family of sets of cardinality at least 2 that contain j, and let $\bar{\mathcal{S}}$ be those that do not contain j. Finally, let $\mathcal{S}_2$ denote the 2-element sets of $\mathcal{S}$.

As above, we know that there is a constant $0 < \ \leq 1$, such that for any subset A of cardinality at least 2, $\Pi(A) = \ \Pi'(A)$. Furthermore, we know how to compute Π (and) from (1). There is a natural 1-1 correspondence f between the sets in $\mathcal{S} - \mathcal{S}_2$ and $\bar{\mathcal{S}}$, by simply deleting j; that is, $f(S)$ maps S to $S - \{j\}$. But then, $\Pi(S) = \frac{1-p_j}{p_j}\Pi(f(S))$, and hence the same relation holds for Π' (provided $|S| \geq 3$). Further, $D_j(S) = D_j(f(S))$, since each is the distance from j to $S - \{j\}$. In other words, the conditional expectation of $D_j(S)$ for sets of size at least 3 containing j is exactly equal to the conditional expectation of $D_j(S)$ for sets not containing j (where both expectations are with respect to the distribution Π').

Proposition 9 *For a random set A drawn according to the distribution Π',*

$$E_A[D_j(A)|(j \in A) \wedge (|A| \geq 3)] \leq E_A[D_j(A)|(j \in A)].$$

Proof. We prove instead that

$$E_A[D_j(A)|(j \in A)] \leq E_A[D_j(A)|(j \in A) \wedge (|A| = 2)],$$

which implies the lemma by the basic properties of conditional expectations.

By definition, $D_j(\{j,k\}) = c(j,k)$. Let σ be a permutation on N such that

$$0 = c(j,j) = c(j,\sigma(1)) \leq c(j,\sigma(2)) \leq \cdots \leq c(j,\sigma(n)).$$

To compute either of these expectations, we need only consider the probability that each conditional distribution takes on each of these $n-1$ non-trivial values. We know that $D_j(A) = c(j,\sigma(k))$ exactly when $\sigma(\ell) \notin A$, $\ell = 2,\ldots,k-1$, and $\sigma(k) \in A$. Furthermore, for each distribution, the probability that any set is selected is proportional to the probability that it was selected in the original distribution Π, for which we know how to compute these probabilities exactly. Thus, for the first expectation, the probability that $D_j(A) = c(j,\sigma(k))$ is proportional to $p_{\sigma(k)} \cdot \prod_{\ell=2,\ldots,k-1}(1-p_{\sigma(\ell)})$, whereas for the second, it is proportional to $p_{\sigma(k)} \cdot \prod_{\ell \neq 1,k}(1-p_{\sigma(\ell)})$ (with the same constant of proportionality). It is clear from these two expressions that the first probability dominates the second, so that for any k, it is more likely in the second case that $D_j(S) \geq c(j,\sigma(k))$. Consequently, the second expectation is at least the first, and the lemma is proved.

It follows that

$$E_A[D_j(A)|j \notin A] \leq E_A[D_j(A)|j \in A].$$

The lemma follows immediately since the unconditioned expectation is sandwiched in between these two values.

There is one final detail that should be mentioned. When we wish to select a sample from Π', the most natural way is to repeatedly choose a sample from Π, until the resulting set has cardinality at least 2. If, for example, each of the values $p_i = 2^{-n}$, then this could take exponential time. However, one can easily compute the conditional probability that the two smallest indexed elements in S are $i < j$ (for any such pair), and then the remaining elements can be selected by the independent selection rule from among $\{j+1, \ldots, n\}$. In this way, we can generate a sample according to Π' in polynomial time. Thus we have derived the following theorem.

Theorem 10. *There is a randomized algorithm that computes a solution τ for the* a priori *TSP in polynomial time and has expected value (with respect to the random choices of the algorithm) no more than four times the optimal value.*

2.4 Derandomization

A stronger result would be to give a deterministic algorithm with the same performance guarantee. In fact, our algorithm in the simpler case with $p_1 = 1$ is essentially identical to an earlier algorithm for the *rent-or-buy problem*, and we will exploit this connection in devising a deterministic analogue of Theorem 10. In this problem, one is given an undirected graph with edge costs c_e and an inflation factor M. One can rent edge e at a cost of c_e per transmission, or buy the edge e (and have unlimited capacity) for a total cost of Mc_e. A given subset of nodes are demand points that need to communicate with the root node 1. The aim is to decide which edges to buy and which to rent so that the total cost is minimized.

Gupta, Kumar, Pál, and Roughgarden [8] gave a very elegant approximation algorithm for this problem, which works as follows: choose a subset S by including node 1, and including each demand node i, independently, with probability $1/M$; build a minimum spanning tree (or a near-optimal Steiner tree) on S, and these are the edges to buy; for each demand node $i \notin S$, rent the shortest edge from i to a node in S to serve that demand. Other than the fact that the inclusion probability for node i is $1/M$, rather than a specified parameter p_i, this is exactly the same algorithm as our approximation algorithm. Furthermore, Williamson and van Zuylen [10] have shown how to deterministically choose a tree for which the cost is at most the expected value of this randomized procedure. In fact, they gave a derandomization procedure for a slight restatement of this problem, called the *connected facility location problem*. In this problem, one selects some facilities to open, and then must connect them by a Steiner tree; for each node j not in the Steiner tree, one connects j to a node in i in the tree, by selecting the shortest such path (though of course, in a setting like ours in which the distances satisfy the triangle inequality, that path would be nothing more than a single edge).

It is quite straightforward to show that the natural generalization of the derandomization approach of Williamson and van Zuylen applies to our setting as well. Consider the bound that we used to analyze the performance of our algorithm, equation (2). This is the upper bound used in the expectation calculation for each active A, and so, for a particular choice of S, the overall bound that we obtain in this way is equal to twice

$$MST(S) + \sum_{j \neq 1} E_A[\mathbb{1}(j \in A)]\mathbb{1}(j \notin S) D_j(S) = MST(S) + \sum_{j \notin S} p_j D_j(S). \quad (8)$$

The essence of our proof is that if we select S at random, then this bound is good relative to the optimal cost. Thus, if we deterministically select a set S for which the actual cost (resulting from this upper bound on the expectation calculation) is no more than this expectation, then we have a deterministic approximation algorithm. It is not difficult to see that the problem of choosing a set S minimizing the expression (8) is an instance of the connected facility location problem, and hence using a (deterministic) 3.28-approximation (from [10]) to the problem gives a deterministic 13.12-approximation to the *a priori* TSP.

We next show how to use ideas from [10] directly to get a deterministic 8-approximation. The standard approach to derandomizing such an algorithm is the so-called *method of conditional expectations*; in this approach, one considers one random choice at a time, such as whether node j is in the set S, and computes the two conditional expectations, with this condition, and with its negation. While the conditional expectation of the second term in (8) is easy to compute, handling the first term seems rather difficult. Instead, Williamson & van Zuylen relied on a linear programming relaxation that captures enough of the structure of the tree optimization problem for which the method of conditional expectations can be applied. Thus, instead of computing the conditional expectation, we compute an upper bound (i.e., a pessimistic estimator) on this conditional expectation.

We first write the standard LP relaxation for the Steiner tree problem on a point set S, where $\delta(U)$ denotes the set of edges ij such that $i \in U$, $j \notin U$:

$$\begin{array}{lll} & \min & \sum_{ij} c(i,j) y_{ij} \\ (\text{Steiner}(S)) & \text{s.t.} & \sum_{ij \in \delta(U)} y_{ij} \geq 1 \ \forall U \subseteq N : S \cap U \neq \emptyset, 1 \notin U \\ & & y_{ij} \geq 0 \end{array}$$

We next write a connected facility location type linear program:

$$\begin{array}{lll} & \min \quad B + C & \\ & \text{s.t.} \quad \sum_j x_{ij} = 1 & \forall i \in N \\ (\text{CFL}) & \sum_{ij \in \delta(U)} z_{ij} \geq \sum_{j \in U} x_{ij} & \forall i \in N, \forall U \subseteq N : 1 \notin U \\ & B = \sum_{ij} c(i,j) z_{ij} & \\ & C = \sum_{i \in N} p_i \sum_{j \in N} c(i,j) x_{ij} & \\ & z_{ij}, x_{ij} \geq 0 & \forall i,j \in N \end{array}$$

Let (x^*, z^*, B^*, C^*) be an optimal solution to (CFL). We prove a sequence of lemmas:

Lemma 11

$$B^* + C^* \leq \frac{3}{2} OPT$$

Proof. Let OPT_A denote an optimal tour for instance A. For a sample A, let z^A be half the incidence vector of the optimal tour on A and for any i, let $x_{ij}^A = 1$ if j is the nearest neighbor of i in $A \setminus \{i\}$. Clearly $B^A = \frac{1}{2} OPT_A$ and $C^A = \sum_i p_i D_j(A)$. Note that the tour OPT_A crosses each set $U \subseteq N : U \cap A \neq 0, 1 \notin U$ at least twice so that the fractional value $\sum_{ij \in \delta(U)} z_{ij}^A$ is at least one. It is then easy to check that (x^A, z^A, B^A, C^A) is a feasible solution for the linear program with objective function value $\frac{1}{2} OPT_A + \sum_j p_j D_j(A)$.

The vector $E_A[(x^A, z^A, B^A, C^A)]$ is a convex combination of feasible points and hence is feasible itself. The objective function value for this solution, using Fact 4 and equation (5) is easily bounded by $\frac{3}{2} OPT$. Thus the optimal solution is no worse.

Given a set $S \subseteq N$, we now define a solution to Steiner(S) as follows:

$$\bar{y}_{ij}^S = z_{ij} + \mathbb{1}(i \in S) x_{ij}.$$

Lemma 12 *$\bar{y}^S$ is a feasible solution to Steiner(S).*

Proof. For any $U \subseteq N : S \cap U \neq \emptyset, 1 \notin U$, let $i' \in S \cap U$. Then:

$$\sum_{kl \in \delta(U)} \bar{y}_{kl}^S = \sum_{kl \in \delta(U)} z_{kl} + \mathbb{1}(k \in S) x_{kl} \geq \sum_{kl \in \delta(U)} z_{kl} + \sum_{l \notin U} x_{i'l} \geq \sum_{l \in U} x_{i'l} + \sum_{l \notin U} x_{i'l} = 1.$$

The claim follows.

With the above lemma in mind, we define $\bar{c}_{ST}(S)$ to be the objective function value of this solution $\bar{y}^S$ for Steiner(S). One can check that the conditional expectation of $\bar{c}_{ST}(S)$ can be directly evaluated. Moreover, since $\bar{y}^S$ is a feasible solution to Steiner(S), one can find a Steiner tree on S with cost at most $2\bar{c}_{ST}(S)$. Further, let $c_R(S)$ denote $\sum_j p_j D_j(S)$.

The method of conditional expectation iterates through the points one by one and decides for each point whether or not to add it to S. We let $P \subseteq N$ denote the points we decide to add to S and let $\bar{P} \subseteq N$ denote the set of points we decide not to add. We define our estimator

$$Est(P, \bar{P}) = 2E[\bar{c}_{ST}(S) | P \subseteq S, \bar{P} \cap S = \emptyset] + E[c_R(S) | P \subseteq S, \bar{P} \cap S = \emptyset].$$

We start with $P, \bar{P}$ empty, and maintain the invariant:

$$Est(P, \bar{P}) \leq 4OPT.$$

This holds in the beginning since $2E[\bar{c}_{ST}(S)]$ is exactly $2(B^* + C^*) \leq 3OPT$ by Lemma 11, and the second term is at most OPT, by Fact 4. Moreover, one can verify (see [10]) that for $i \notin P, \bar{P}$,

$$Est(P, \bar{P}) = p_i Est(P \cup \{i\}, \bar{P}) + (1 - p_i) Est(P, \bar{P} \cup \{i\}).$$

Thus the smaller of $Est(P \cup \{i\}, \bar{P})$ and $Est(P, \bar{P} \cup \{i\})$ is no larger than $Est(P, \bar{P})$. We can therefore add i to one of P or $\bar{P}$ without violating the invariant.

At the end of the process, $P \cup \bar{P} = N$ and we have a deterministic set S for which $2\bar{c}_{ST}(S) + c_R(S) \leq 4OPT$. Doubling the tree costs us another factor of two, and so we get an 8-approximation algorithm.

We have explained the derandomization in the context of the rooted variant. However, for the deterministic algorithm the distinction between these two cases essentially disappears. The probabilistic argument used to analyze the theorem proves that the expected cost over sets of size at least two is good (and can be bounded in the same way as previously). However, we can simply try all possible choices for the lexicographically smallest pair of nodes in S, choose the first as the root, and then continue the derandomization procedure for the remaining nodes. For each of the resulting at most n^2 choices of S, we can evaluate the upper bound (8), and choose the best. The previous theorem ensures that the resulting solution will have the desired guarantee.

Theorem 13. *There is a deterministic algorithm that computes a solution τ for the* a priori *TSP in polynomial time and has objective function value no more than eight times the optimal value.*

3 Conclusions

The constant 4 (or 8 for the deterministic case) can probably be optimized. There are at least two immediate possible sources of improvement. First, we are using the "double-tree" heuristic to compute the tour, rather than Christofides's algorithm. This requires a slightly different analysis (since one must also bound the cost of the matching in terms of the *a priori* TSP). Furthermore, unlike in the result of Williamson & van Zuylen for rent-or-buy, we lose an extra factor of two relative to an LP relaxation. It seems likely that by adapting this technique to a stronger LP relaxation one can avoid this loss. On the other hand, our algorithm is competitive with respect to the expected value of the ex post OPT, which is a lower bound on the optimal cost for the problem. It is conceivable that one can get a better approximation guarantee if one uses a better lower bound on OPT. Finally, it is important to note that this algorithm is not a constant approximation algorithm in the black box model (or even the polynomial scenario model) for representing the probability distribution Π. It remains an interesting open question to provide a constant approximation algorithm in either of those two settings.

Acknowledgments The authors would like to thank Alan Frieze, Shane Henderson, Frans Schalekamp, Anke van Zuylen, and David Williamson for helpful discussions.

References

1. Jaillet, P.: Probabilistic traveling salesman problems. Technical Report 185, Operations Research Center, MIT (1985)
2. Jaillet, P.: A priori solution of a traveling salesman problem in which a random subset of the customers are visited. Operations Research **36** (1988) 929–936
3. Bertsimas, D.: Probabilistic Combinatorial Optimization Problems. PhD thesis, MIT, Cambridge, Mass. (1988)
4. Bertsimas, D.J., Jaillet, P., Odoni, A.R.: A priori optimization. Operations Research **38**(6) (1990) 1019–1033
5. Schalekamp, F., Shmoys, D.B.: Algorithms for the universal and *a priori* TSP. Operations Research Letters (2007) In press.
6. Bartholdi, III, J.J., Platzman, L.K.: An $O(N \log N)$ planar travelling salesman heuristic based on spacefilling curves. Operations Research Letters **1**(4) (1981/82) 121–125
7. Hajiaghayi, M.T., Kleinberg, R., Leighton, T.: Improved lower and upper bounds for universal tsp in planar metrics. In: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, New York, NY, USA, ACM Press (2006) 649–658
8. Gupta, A., Kumar, A., Pál, M., Roughgarden, T.: Approximation via cost-sharing: a simple approximation algorithm for the multicommodity rent-or-buy problem. In: Proceedings of the 44th Anuual IEEE Symposium on Foundations of Computer Science, Silver Spring, MD, IEEE Computer Society (2003) 606–615
9. Gupta, A., Pál, M., Ravi, R., Sinha, A.: Boosted sampling: approximation algorithms for stochastic optimization. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing. (2004) 265–274
10. Williamson, D.P., van Zuylen, A.: A simpler and better derandomization of an approximation algorithm for single source rent-or-buy. Operations Research Letters **35** (2007) 707–712
11. Garg, N., Gupta, A., Leonardi, S., Sankowski, P. Private communication (2007)
12. Garg, N., Gupta, A., Leonardi, S., Sankowski, P.: Stochastic analyses for online combinatorial optimization problems. In: Proceedings of the 19th Anuual ACM-SIAM Symposium on Discrete Algorithms, NY and Philadelphia, ACM and SIAM (2008) 942–951
13. Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B., eds.: The Traveling Salesman Problem. Wiley (1985)