Algorithms and Adaptivity Gaps for Stochastic k-TSP

Haotian Jiang
Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, USA
jhtdavid@cs.washington.edu

Jian Li
Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
lijian83@mail.tsinghua.edu.cn

Daogao Liu
Department of Physics, Tsinghua University, Beijing, China
liudg16@mails.tsinghua.edu.cn

Sahil Singla
Princeton University and Institute for Advanced Study, Princeton, USA
singla@cs.princeton.edu

Abstract

Given a metric (V, d) and a root $\in V$, the classic k-TSP problem is to find a tour originating at the root of minimum length that visits at least k nodes in V. In this work, motivated by applications where the input to an optimization problem is uncertain, we study two stochastic versions of k-TSP.

In Stoch-Reward k-TSP, originally defined by Ene-Nagarajan-Saket [13], each vertex v in the given metric (V, d) contains a stochastic reward R_v. The goal is to adaptively find a tour of minimum *expected* length that collects at least reward k; here "adaptively" means our next decision may depend on previous outcomes. Ene et al. give an $O(\log k)$-approximation adaptive algorithm for this problem, and left open if there is an $O(1)$-approximation algorithm. We totally resolve their open question, and even give an $O(1)$-approximation *non-adaptive* algorithm for Stoch-Reward k-TSP.

We also introduce and obtain similar results for the Stoch-Cost k-TSP problem. In this problem each vertex v has a stochastic cost C_v, and the goal is to visit and select at least k vertices to minimize the expected *sum* of tour length and cost of selected vertices. Besides being a natural stochastic generalization of k-TSP, this problem is also interesting because it generalizes the Price of Information framework [33] from deterministic probing costs to metric probing costs.

Our techniques are based on two crucial ideas: "repetitions" and "critical scaling". In general, replacing a random variable with its expectation leads to very poor results. We show that for our problems, if we truncate the random variables at an ideal threshold, then their expected values form a good surrogate. Here, we rely on running several repetitions of our algorithm with the same threshold, and then argue concentration using Freedman's and Jogdeo-Samuels' inequalities. Unfortunately, this ideal threshold depends on how far we are from achieving our target k, which a non-adaptive algorithm does not know. To overcome this barrier, we truncate the random variables at various different scales and identify a "critical" scale.

2012 ACM Subject Classification Theory of computation → Stochastic control and optimization; Theory of computation → Approximation algorithms analysis

Keywords and phrases approximation algorithms, stochastic optimization, travelling salesman problem

Digital Object Identifier 10.4230/LIPIcs.ITCS.2020.45

Acknowledgements We are thankful to the anonymous reviewers of ITCS 2020 for many helpful comments on the presentation of this paper. Haotian Jiang is supported in part by NSF awards CCF-1749609, CCF-1740551 and DMS-1839116. Jian Li and Daogao Liu are supported in part by the National Natural Science Foundation of China Grant 61822203, 61772297, 61632016, 61761146003, and the Zhongguancun Haihua Institute for Frontier Information Technology and Turing AI Institute of Nanjing. Sahil Singla is supported in part by the Schmidt Foundation.

1 Introduction

Consider a scenario where a salesperson must sell some quota of brushes in order to win a trip to Hawaii. The salesperson knows the time it takes to travel between different cities and the demand at each city. What is the best route to take to sell the quota while spending the least amount of time? This exact scenario was described by Awerbuch et al. [4] to motivate the study of TSP problems where the algorithm has to also decide which cities to visit. A cleaner version of this problem, first introduced by Ravi et al. [31], is the k-TSP problem where we assume that each city has a unit demand. Formally, given a metric (V, d) with a root $\in V$ and a *target* $k \in \mathbb{Z}_{\geq 0}$, the k-TSP problem is to find a tour that originates at the root and visits at least k vertices, while minimizing the total travel time. There is a long line of work trying to design better approximation algorithms for the k-TSP problem [4, 30, 8, 16, 2, 1], and the state-of-the-art is a 2-approximation algorithm due to Garg [17][1].

In this work we consider stochastic versions of the k-TSP problem: what if the salesperson does not know the exact demand in each city, or what if the salesperson need to spend some uncertain time in each city to complete the city's demand? Indeed, there is a long line of work studying classical optimization problems where we begin only with estimates (probability distributions) on the input parameters. The algorithm has to *adaptively probe* parameters (inspect elements) by paying some "cost" before realizing their exact values; here "adaptively" means that our decisions may depend on the outcomes of already probed elements. Such stochastic probing problems have been well-studied in both maximization and minimization settings [11, 18, 19, 22, 20, 23, 5, 24, 6, 29, 3, 25, 13, 15, 33, 9, 21].

There are two natural ways of defining the stochastic k-TSP problem, depending on the type of input uncertainty. In the Stoch-Reward k-TSP problem, first introduced by Ene-Nagarajan-Saket [13], we incorporate uncertainty in the vertex demands. Formally, we assume that the demand at each vertex is drawn independently from a known distribution, and the goal is to *adaptively* find a tour Π that obtains the target demand k, while minimizing the total *expected* travel time[2].

We also study Stoch-Cost k-TSP where each city still has a unit demand, but the salesperson will have to spend an additional *completion* time (drawn from a known distribution) at each vertex before meeting its unit demand. The goal is to adaptively find a tour Π that completes the target demand k, while minimizing the expected *sum* of total travel and completion times. Notice, our algorithm finds the exact completion time of a vertex only after visiting it, and may choose not to complete it (i.e., not meet its unit demand) if the completion time seems too long. This idea of studying stochastic completion times at vertices is not new, and has been previously used in the study of stochastic Orienteering problems, which in some sense is the dual to our Stoch-Cost k-TSP problem [23, 6].

A common theme in the study of stochastic probing problems is to understand the power of adaptivity. Indeed, while the optimal algorithms can fully adapt to the outcomes, and hence may not even have a polynomial-size representation, a *non-adaptive* algorithm makes all its decisions upfront independent of the observed outcomes (except perhaps the stopping time). Being non-adaptive has several benefits like they are simpler to find, easily parallelizable, and have a poly-size representation. So ideally for a probing problem we would like to design non-adaptive algorithms with performance close to the optimal adaptive

[1] A closely-related variant is called the k-MST problem. Both problems are equivalent up to a constant approximation factor.
[2] Our Stoch-Reward k-TSP problem is called the "Stochastic k-TSP" problem in [13]. We rename it to differentiate it from Stoch-Cost k-TSP.

algorithms, or in other words design non-adaptive algorithms with a small *adaptivity gap*. The main results of our work is to show that both the above Stoch-Reward k-TSP and Stoch-Cost k-TSP problems have a constant adaptivity-gap. That is, there exist fixed-tours starting at the root which the algorithm can take until it obtains the target demand k, which guarantee an expected total time at most a constant factor more than the expected total time of the optimal adaptive tour. Moreover, for distributions with polynomial support, we give poly-time algorithms to find such tours.

Our constant adaptivity-gap for Stoch-Reward k-TSP answers the main open question of [13], who showed an $O(\log^2 k)$ bound on the adaptivity gap. The constant adaptivity gap result for Stoch-Cost k-TSP might also seem surprising because it is known that the related Stochastic Orienteering problem has a super-constant adaptivity gap [6].

In the rest of this section we first formally state our problems and results, and then discuss our high-level techniques and other related work.

1.1 Stoch-Reward k-TSP

The following Stoch-Reward k-TSP was first defined by Ene et al. [13].

Stoch-Reward k-TSP. We are given a metric (V, d) with a root $\in V$ and each vertex $v \in V$ has an independent integral[3] stochastic[4] reward $R_v \in \mathbb{Z}_{\geq 0}$. All reward distributions are given as input but the actual reward instantiation R_v is only known when the algorithm visits vertex v. Given a target value $k \in \mathbb{Z}_{\geq 0}$, the goal is to adaptively find a tour Π originating at root that collects at least k reward (i.e., $\sum_{v\in\Pi} R_v \geq k$) while minimizing the expected tour length.

This problem captures several well-studied problems; e.g., it captures the problem of Stoch-Knapsack Cover where the metric (V, d) is a weighted star: given a target k and n items where item $i \in [n]$ has both a deterministic cost $C_i \in \mathbb{R}_{\geq 0}$ and an independent stochastic reward $R_i \in \mathbb{Z}_{\geq 0}$, the Stoch-Knapsack Cover problem is to adaptively obtain a total reward of at least k at the minimum expected cost. This problem was studied by Deshpande et al. [12], and they gave an *adaptive* 2-approximation algorithm. However, even in this special case, it was not know if there is a non-adaptive constant factor approximation algorithm.

The first non-trivial results for the Stoch-Reward k-TSP problem were obtained by Ene et al. [13]. They gave an $O(\log^2 k)$-approximation non-adaptive algorithm and an $O(\log k)$-approximation adaptive algorithm. On the hardness side, however, they only gave a lower bound of e on the adaptivity gap. This left open closing the wide gap on the adaptivity gap for Stoch-Reward k-TSP. We resolve their open question by giving a non-adaptive $O(1)$-approximation algorithm.

▶ **Theorem 1.** *The Stoch-Reward k-TSP problem has a non-adaptive $O(1)$-approximation algorithm.*

The difficulty in Stoch-Reward k-TSP arises because the expected reward is a poor indicator of how much we care about a node. An extreme example is a vertex with a large expected reward, but which is non-zero with nearly zero probability. It is therefore reasonable to truncate the reward distributions at the remaining target reward. However, it is not clear why such an approach would work, and moreover this approach is adaptive as it depends on the remaining target.

[3] This is without loss of generality. Our result also generalizes to the case where rewards are real numbers via re-scaling.
[4] We assume that the distribution is discrete and is given explicitly.

1.2 Stoch-Cost k-TSP

We formally define the Stoch-Cost k-TSP problem.

Stoch-Cost k-TSP. We are given a metric (V, d) with a $\mathsf{root} \in V$ and each vertex $v \in V$ has an independent stochastic cost $C_v \in \mathbb{R}_{\geq 0}$. All cost distributions are given as input but the actual cost instantiation C_v is only known when vertex v is visited. Suppose a vertex v can only be *selected* if: (1) v is visited and (2) we are currently[5] at vertex v. The goal is to adaptively find a tour Π originating at root that selects a set S of k visited vertices while minimizing the expected *total* cost, which is the sum of the tour length and the cost of the selected vertices:

$$\mathbb{E}\Big[d(\Pi) + \sum_{v \in S} C_v\Big].$$

Apart from being a natural generalization of the classical k-TSP problem, Stoch-Cost k-TSP is also motivated due to its connections to price of information [33]. In particular, it generalizes the Minimization k-Pandora's Box problem studied in [28, 33]. In this problem we are given a target k and n items, where each item $i \in [n]$ has a known probing price $\pi_i \in \mathbb{R}_{\geq 0}$ and an independent stochastic cost C_i. The exact cost C_i is only revealed after we pay price π_i. The goal is to adaptively probe and select k of the probed items to minimize the expected total selection cost plus probing price. Stoch-Cost k-TSP captures this problem on a star metric where node i is at a distance $\pi_i/2$ from the root. Thus, we can view the Stoch-Cost k-TSP problem as generalizing the price of information framework to a *metric* setting, where the price of probing is not fixed but given by a general metric. Our next result gives a non-adaptive $O(1)$-approximation algorithm for Stoch-Cost k-TSP.

▶ **Theorem 2.** *The Stoch-Cost k-TSP problem has a non-adaptive $O(1)$-approximation algorithm.*

Our main techniques in the proof of Theorem 2 are similar to those for Stoch-Reward k-TSP. In fact, in Section 3 we present a generic framework that can be used to solve both these problems. It will be interesting to find other applications of our framework in future work.

1.3 High-level Techniques

We assume that after re-scaling, the distance between any pair of vertices is at least 1.

Stoch-Reward k-TSP. A standard idea in the design of approximation algorithms on a metric is to operate in phases, where in phase i our algorithm is allowed a budget of 2^i. Intuitively, this corresponds to the algorithm imagining that the optimal adaptive tour has length $\Theta(2^i)$. In each phase, a naïve algorithm would be to collect as much *expected* reward as possible within budget 2^i (say, by solving an instance of the Orienteering problem). However, in general the performance of such a naïve algorithm can be arbitrarily bad. E.g., suppose $k - \sqrt{k}$ reward is easy to get, now for the remaining reward the naïve algorithm prefers a vertex having a reward of k with probability $10/\sqrt{k}$ and 0 otherwise, as opposed to a vertex with a deterministic reward of $\sqrt{k}$ (assuming both are at the same distance).

[5] Up to a factor of 2, this version of the problem is equivalent to the version where restriction (2) is removed.

A natural fix to the above issue is to *truncate* the reward distributions at the *remaining* target reward and then take expectations. Not only is this algorithm *adaptive*, it is not even clear why it would work. Indeed, Ene et al. [13] give an example (see Example 2 in [13]) where this algorithm has an $\Omega(\log k)$-approximation factor. Our first idea is to run $O(1)$ *repetitions* of a bi-criteria Orienteering algorithm using the *same* truncation (i.e., the initial remaining target) for all these repetitions. Our analysis applies Freedman's [14] and Jogdeo-Samuels [27] inequalities to argue concentration, and relies crucially on not updating the remaining target for these $O(1)$ repetitions.

Nevertheless, the above approach depends on the remaining target, which is unknown to a *non-adaptive* algorithm. One could bypass this by truncating the reward distributions at $\log k$ different *scales*, where scale j corresponds to the remaining target being roughly $k/2^j$, applying the previous $O(1)$ repetitions idea at each scale, and visiting the union of all tours. Unfortunately, this immediately loses a $\log k$ approximation factor. Our second idea is "critical scaling" in which we identify a "critical" scale j_{crit} among the $\log k$ possible scales, and only include tours for scales $j_{\mathsf{crit}} - 1$ and j_{crit}. This critical scale j_{crit} roughly (but not quite) corresponds to a "phase transition" from an underestimation to an overestimation of the remaining target. A priori it is not clear why such a "critical" scale can be found non-adaptively, but the concentration properties of the above $O(1)$ repetitions allow us to find it efficiently.

Stoch-Cost k-TSP. We obtain our result for Stoch-Cost k-TSP in a similar way. An immediate challenge, however, is how should we truncate the cost distributions C_v, say even if the remaining target k' is known? One natural way is by looking at $\mathbb{P}[C_v \le O(2^i/k')]$, where $O(2^i/k')$ is the "average" cost per remaining reward in phase i. But such an approach would fail when some vertices in the optimal tour have costs much smaller than $O(2^i/k')$, while the other vertices have much higher costs. We overcome this by considering $\mathbb{P}[C_v \le O(2^{i-j})]$ for all possible scales $j \in \{0, \cdots, i + \log n\}$, identifying a "critical" scale j_{crit}, and again only including the tours for scales $j_{\mathsf{crit}} - 1$ and j_{crit}. To identify the critical scale, we need to evaluate the maximum target a tour at a given scale can get within cost budget 2^i with constant probability. We show this can be approximately computed via dynamic programming.

1.4 Further Related Work

There is a long line of work studying the classic k-TSP and the related k-MST problem; we refer the readers to Garg's beautiful 2-approximation paper and the references therein [17].

A formal study on the benefits of adaptivity for stochastic combinatorial optimization problems started with the seminal work of Dean et al. [11]. They showed that for the stochastic knapsack problem, where items sizes are independently drawn and we need to fit them in a knapsack of size B, there is an $O(1)$-approximation non-adaptive algorithm. This factor was later improved to a $(2 + \epsilon)$-approximation in [7, 29]. The minimization version of the stochastic knapsack problem, which is known as the Stoch-Knapsack Cover problem, was studied by Deshpande et al. [12], and is a special case of Stoch-Reward k-TSP as we mentioned before. The unbounded version of Stoch-Knapsack Cover (each item has infinite number of copies) was studied by [26] and they provide an FPTAS for this problem.

Gupta et al. [23] generalized the stochastic knapsack problem to the stochastic orienteering problem, where each stochastic item now resides on a vertex of a given metric, and we need to fit both the tour length and the item sizes inside our budget B. They give an $O(\log \log B)$-approximation non-adaptive algorithm for this problem. Bansal and Nagarajan [6] later showed that this problem has no constant-approximation non-adaptive algorithm. These works inspired Ene et al. [13] to study Stoch-Reward k-TSP, a natural minimization variant of the stochastic orienteering problem. Prior to our work, it was conceivable that this minimization problem also has a super-constant adaptivity gap, like stochastic orienteering.

Motivated by different applications that solve discrete problems under an uncertain input, other related stochastic probing models have been studied. We refer the readers to Singla's Ph.D. Thesis for a survey [32]. Of particular interest to us is the Price of Information model [33], which was inspired from the work on Pandora's box [34, 28]. Their Minimization k-Pandora's Box problem inspired us to define Stoch-Cost k-TSP, which generalizes the probing costs from being fixed to being on a metric. Although an optimal strategy is known for Minimization k-Pandora's Box, the problem becomes APX-hard on a metric as it generalizes k-TSP.

Organization

We start with some preliminary definitions and lemmas in Section 2. In Section 3, we describe our general framework for both Stoch-Reward k-TSP and Stoch-Cost k-TSP, and prove some key lemmas that will be used throughout the paper. We give our non-adaptive $O(1)$-approximation algorithm for Stoch-Reward k-TSP that proves Theorem 1 in Section 4. The non-adaptive $O(1)$-approximation algorithm for Stoch-Cost k-TSP that proves Theorem 2 is in Section 5.

2 Preliminaries

2.1 Adaptive vs Non-Adaptive Algorithms

Any feasible solution to our stochastic problems can be described by a *decision tree*, where nodes correspond to vertices that are visited and branches correspond to instantiations of the observed random variables. Even if the degree of every vertex is a constant, the size of such decision trees can be exponentially large in its height. These solutions are called *adaptive* because the choice of the next vertex to visit depends on the outcomes of the already visited nodes.

We also consider the special class of *non-adaptive* solutions that is described simply by an ordered list of vertices: the policy involves visiting vertices in the given order until a certain stopping criterion is met. Such non-adaptive solutions are often preferred over adaptive solutions because they are easier to implement.

In this work we only study minimization problems. We compare the performance of our algorithm with that of the optimal *adaptive* algorithm, which is denoted by OPT. We abuse notation and also use OPT to denote the expected objective of the optimal adaptive algorithm. We say an algorithm is α-approximation for $\alpha \geq 1$ if the expected objective of the algorithm is at most $\alpha \cdot \mathsf{OPT}$. Ideally, we want to design non-adaptive algorithms whose performance is comparable to the optimal adaptive algorithm. Since this is not always possible, it is important to bound the *adaptivity gap*, which is the worst-case ratio between the expected objectives of the optimal non-adaptive and the optimal adaptive algorithms.

2.2 Probability Inequalities

Our proofs will require the following probability inequalities. We start with a bound on the median of independent Bernoulli random variables due to Jogdeo and Samuels [27]. Given n independent Bernoulli random variables $X_1, \cdots, X_n$ where X_i has success probability $p_i \in [0,1]$, let $X := \sum_{i=1}^{n} X_i$ be their sum. Define the median of X to be any integer $m \in \mathbb{Z}_{\geq 0}$ such that $\min\{\mathbb{P}[X \geq m], \mathbb{P}[X \leq m]\} \geq 1/2$.

▶ **Theorem 3** (Theorem 3.2 and Corollary 3.1 [27]). *Let $X = \sum_{i=1}^{n} X_i$ be the sum of n independent Bernoulli random variables where X_i has success probability $p_i \in [0,1]$. If $\mathbb{E}[X]$ is an integer k, then the median of X is also k. If $k < \mathbb{E}[X] < k+1$ for some integer k, then the median of X is either k or $k+1$.*

We will also need the following martingale inequality due to Freedman [14].

▶ **Theorem 4** (Freedman's Inequality, Theorem 1.6 in [14]). *Consider a real-valued martingale sequence $\{X_t\}_{t\geq 0}$ such that $X_0 = 0$, and $\mathbb{E}[X_{t+1}|\mathcal{F}_t] = 0$ for all t, where $\{\mathcal{F}_t\}_{t\geq 0}$ is the filtration defined by the martingale. Assume that the sequence is uniformly bounded, i.e., $|X_t| \leq M$ almost surely for all t. Now define the predictable quadratic variation process of the martingale to be $W_t = \sum_{j=1}^{t} \mathbb{E}[X_j^2|\mathcal{F}_{j-1}]$ for all $t \geq 1$. Then for all $\ell \geq 0$ and $\sigma^2 > 0$ and any stopping time τ, we have*

$$\mathbb{P}\Big[\Big|\sum_{j=0}^{\tau} X_j\Big| \geq \ell \wedge W_\tau \leq \sigma^2 \textit{for some stopping time } \tau\Big] \leq 2\exp\Big(-\frac{\ell^2/2}{\sigma^2 + M\ell/3}\Big).$$

▶ **Theorem 5** (Chernoff Bound). *Let $X_1, X_2, \cdots, X_n$ be independent random variables taking values in $[0,1]$ and define $X := \sum_{i\in[n]} X_i$. Then for any $\delta \in [0,1]$, we have*

$$\mathbb{P}\left[X \leq (1-\delta)\cdot\mathbb{E}[X]\right] \leq \exp\left(-\delta^2 \cdot \mathbb{E}[X]/2\right).$$

2.3 A Bi-Criteria Algorithm $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ for Orienteering

We formally define the well-known Orienteering problem.

Orienteering. Given a metric (V, d) with root $\in V$, a *profit*[6] $R_v > 0$ for each $v \in V$, and a budget $B > 0$, the goal is to find a tour originating at root of length at most B that maximizes the collected profit.

The state-of-the-art for this NP-hard Orienteering problem is a $(2+\epsilon)$-approximation algorithm [10]. We denote this algorithm as $\mathsf{ALG}_{\mathsf{Orient}}$ and denote the profit of the optimal Orienteering tour as $\mathsf{OPT}_{\mathsf{Orient}}$. For our purposes, however, we also need to find profit at least $\mathsf{OPT}_{\mathsf{Orient}}$ minus an arbitrarily small additive error. To achieve this, the tour found by our algorithm has length $O(1)\cdot B$.

▶ **Lemma 6** (Bi-criteria Orienteering). *There is an efficient algorithm $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ that finds a tour of length $O(1)\cdot B$ while collecting at least $(\mathsf{OPT}_{\mathsf{Orient}} - \epsilon)$ profit, where $\epsilon = 1/\mathrm{poly}(n)$ can be made arbitrarily small.*

3 Our Approach via Critical Scaling and Repetitions

3.1 A Meta-Algorithm and Critical Scaling

Our non-adaptive $O(1)$-approximation algorithms for both the problems, Stoch-Reward k-TSP and Stoch-Cost k-TSP, have the same structure described in Meta-Algorithm $\mathsf{ALG}_{\mathsf{Meta}}$ (Algorithm 1). This algorithm operates in phases where it gets a budget of $O(1)\cdot\gamma^i$ in phase $i \geq 0$ for some constant $\gamma \in (1,2)$.

[6] We use the word "profit" for Orienteering to avoid confusion with the "reward" in Stoch-Reward k-TSP.

In each phase i, $\mathsf{ALG}_{\mathsf{Meta}}$ explores multiple different "scales" in the remaining graph after excluding the set Π of vertices found in the previous phases. Recall, each scale corresponds to truncating the random variables at a different threshold. This is crucial because our non-adaptive algorithm doesn't know the remaining reward to reach the target k. For each different scale, $\mathsf{ALG}_{\mathsf{Meta}}$ obtains a tour of length $O(1) \cdot \gamma^i$ via a sub-procedure $\mathsf{ALG}_{\mathsf{Rep}}$ to which it sends the truncated random variables as arguments (discussed in Section 3.2). Eventually, $\mathsf{ALG}_{\mathsf{Meta}}$ identifies a "critical" scale j_{crit} and appends at the end of Π the two tours corresponding to the scales j_{crit} and $j_{\mathsf{crit}} - 1$. Since we only append two tours, $\mathsf{ALG}_{\mathsf{Meta}}$ uses budget at most $O(1) \cdot \gamma^i$ in phase i.

Algorithm 1 A Meta-Algorithm $\mathsf{ALG}_{\mathsf{Meta}}$.

```
1  Pre-processing stage:
2  set $\gamma \in (1,2), \Pi \leftarrow \emptyset$ and $\ell \leftarrow \mathsf{polylog}(k,n)$;
3  for phase $i = 0, 1, \cdots$ do
4      set $\Pi_{i,-1} \leftarrow \emptyset$;
5      for scale $j = 0, \cdots, \ell$ do
6          set $X_v^j \in [0,1]$ to be the truncation of $X_v$ at scale $j$ for $v \in V \setminus \Pi$, and zero for
            $v \in \Pi$ ;
7          find tour $\Pi_{i,j} \leftarrow \mathsf{ALG}_{\mathsf{Rep}}\big(\{X_v^j\}_{v \in V \setminus \Pi}\ ,\ i\big)$ of length $O(1) \cdot \gamma^i$;
8      end
9      identify a "critical" scale $j_{\mathsf{crit}}$ and set $\Pi_i \leftarrow \Pi_{i,j_{\mathsf{crit}}} \cup \Pi_{i,j_{\mathsf{crit}}-1}$;
10     append tour $\Pi_i$ to $\Pi$, i.e., $\Pi \leftarrow \Pi \circ \Pi_i$;
11 end
12
13 Probing stage:
14 for phase $i = 0, 1, \cdots$ do
15     visit vertices in the order of $\Pi_i$ and apply certain Selection and Stopping Criteria;
16 end
17 Return set of vertices selected
```

To analyze the algorithm, we need some notation for any phases $i, i' \geq 1$:

- σ_{i-1}: outcome of vertices visited by $\mathsf{ALG}_{\mathsf{Meta}}$'s in the first $i-1$ phases of the probing stage.
- $u_{i'}(\sigma_{i-1})$: probability that $\mathsf{ALG}_{\mathsf{Meta}}$ enters phase $i'+1$ in the probing stage, conditioning on σ_{i-1}.
- $u^*_{i'}(\sigma_{i-1})$: probability that the cost of OPT is more than $\gamma^{i'}$, conditioning on σ_{i-1}.

Notice $u_{i-1}(\sigma_{i-1})$ denotes the indicator variable that $\mathsf{ALG}_{\mathsf{Meta}}$ enters phase i in the probing stage. The following Lemma 7 is the key to our theorems. Roughly, it says that $\mathsf{ALG}_{\mathsf{Meta}}$ is a constant approximation algorithm if it can ensure that whenever $u^*_i(\sigma_{i-1})$ is small (i.e., OPT has a large success probability within budget γ^i) then $\mathsf{ALG}_{\mathsf{Meta}}$ also succeeds with a constant probability in the first i phases (i.e., only using $O(1) \cdot \gamma^i$ budget). The proof of Lemma 7 is standard (e.g., [13]), and we defer it to Appendix B.

▶ **Lemma 7** (Key Lemma). *If for some universal constants $C > 0$, $\gamma > 1$, any phase $i \geq 1$, and any possible σ_{i-1}, the algorithm $\mathsf{ALG}_{\mathsf{Meta}}$ satisfies*

$$u_i(\sigma_{i-1}) \leq C \cdot u^*_i(\sigma_{i-1}) + \frac{u_{i-1}(\sigma_{i-1})}{\gamma^2},$$

then $\mathsf{ALG}_{\mathsf{Meta}}$ is a non-adaptive $O(1)$-approximation algorithm.

All our effort will go in designing $\mathsf{ALG}_{\mathsf{Meta}}$ that satisfies the precondition of Lemma 7.

3.2 $\mathsf{ALG}_{\mathsf{Rep}}$: Constant Repetitions of $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ Suffice

For a fixed scale j in phase i, $\mathsf{ALG}_{\mathsf{Meta}}$ uses $\mathsf{ALG}_{\mathsf{Rep}}$ (Algorithm 2) as a key sub-procedure to find a tour of length $O(1) \cdot \gamma^i$. To achieve this, $\mathsf{ALG}_{\mathsf{Rep}}$ runs a constant number of repetitions of $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ on an Orienteering instance where each vertex v has a profit $w_v = \mathbb{E}[X_v]$ for input random variable $X_v \in [0,1]$ (recall, X_v is the truncated random variable at scale j for vertices outside Π). In each repetition, $\mathsf{ALG}_{\mathsf{Rep}}$ excludes vertices found in previous repetitions.

Algorithm 2 $\mathsf{ALG}_{\mathsf{Rep}}\big(\{X_v\}_{v\in V}\ ,\ i\big)$.

```
1  Input: random variables $X_v \in [0,1]$ corresponding to vertex profits and phase $i$;
2  Main stage:
3  set $\gamma \in (1,2)$, $\epsilon \leftarrow 1/10^5$, $C \leftarrow O(1)$, and $w_v = \mathbb{E}[X_v]$;
4  set $\Pi_i \leftarrow \emptyset$;
5  for repetition $s = 1, \cdots, C$ do
6      use $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ to find a tour $\pi_s$ with budget $\gamma^i$, profit $\{w_v\}_{v\in V}$, and error $\epsilon$;
7      append tour $\pi_s$ to $\Pi_i$, i.e., $\Pi_i \leftarrow \Pi_i \circ \pi_s$;
8      reset $w_v = 0$ for $v \in \Pi_i$;
9  end
10 Return $\Pi_i$
```

Intuition. In the following, we prove two important properties of $\mathsf{ALG}_{\mathsf{Rep}}$. Recall from Algorithm 2 that Π_i denotes the union of the C repetitions of $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$. The first property (Lemma 8) roughly says that if an Orienteering tour of budget γ^i cannot obtain much profit outside Π_i, then OPT also cannot obtain much reward outside Π_i within budget γ^i. The second property (Lemma 9) roughly says that if on the other hand lots of profit can be found by an Orienteering tour outside Π_i, then the tour Π_i obtains a large amount of *expected* reward in its C repetitions, much more than what OPT obtains within budget γ^i. This follows from the property that $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ obtains profit close to the optimal Orienteering tour.

To formally state the above two properties, we need some notation. Consider the Orienteering instance in the remaining graph $V \setminus \Pi_i$ where the budget is γ^i and each vertex $v \in V \setminus \Pi_i$ has profit $\mathbb{E}[X_v]$. Denote $\pi \subseteq V \setminus \Pi_i$ the optimal Orienteering tour for this instance and let

$$T := \sum_{v\in\pi} \mathbb{E}[X_v] \tag{1}$$

be the Orienteering profit obtained by π. For any adaptive strategy ADAP, let $\Pi_i(\mathsf{ADAP}) \subseteq \Pi_i$ denote the random set of vertices visited by ADAP inside the tour Π_i and let $\overline{\Pi}_i(\mathsf{ADAP}) \subseteq V \setminus \Pi_i$ denote the random set of vertices visited by ADAP outside the tour Π_i.

▶ **Lemma 8.** *Suppose we are given independent random variables $X_v \in [0,1]$. Let T be as defined in (1). Then for any adaptive strategy* ADAP *which uses at most γ^i budget and any constant $\alpha > 1$, we have*

$$\mathbb{P}\Big[\sum_{v\in\overline{\Pi}_i(\mathsf{ADAP})} X_v \geq \alpha T\Big] \;\leq\; 2 \cdot \exp\left(-\frac{(\alpha-1)^2 T/2}{1+(\alpha-1)/3}\right).$$

Proof of Lemma 8. We construct a martingale for the (random) set $\overline{\Pi}_i(\mathsf{ADAP})$ of vertices visited by ADAP in $V \setminus \Pi_i$ as follows: When ADAP visits a vertex $v \in \overline{\Pi}_i(\mathsf{ADAP})$, the martingale proceeds for one step with martingale difference defined by

$$Z_v \;:=\; X_v - \mathbb{E}[X_v] \;\in\; [-1,1],$$

and the martingale doesn't move when ADAP visits a vertex $v \in \Pi_i$. The stopping time τ is naturally defined as the martingale step when ADAP finishes. Since each $X_v \in [0,1]$, the quadratic variance of the above martingale $W_\tau = \sum_{j=1}^{\tau} \mathbb{E}[X_j^2|\mathcal{F}_{j-1}]$ is bounded by its expectation $\sum_{j=1}^{\tau} \mathbb{E}[X_j|\mathcal{F}_{j-1}]$ which is at most T, i.e., $\sum_{v\in\overline{\Pi}_i(\mathsf{ADAP})} \mathbb{E}[X_v] \le T$. Therefore, applying Freedman's inequality (Theorem 4), we have

$$\begin{aligned}\mathbb{P}\Big[\sum_{v\in\overline{\Pi}_i(\mathsf{ADAP})} X_v \ge \alpha T\Big] \;&\le\; \mathbb{P}\Big[\big|\sum_{v\in\overline{\Pi}_i(\mathsf{ADAP})} Z_v\big| \ge (\alpha-1)T \wedge W_\tau \le T\Big]\\ &\le\; 2\cdot\exp\left(-\frac{(\alpha-1)^2T/2}{1+(\alpha-1)/3}\right).\end{aligned}$$

This finishes the proof of Lemma 8. ◀

▶ **Lemma 9.** *Suppose we are given independent random variables $X_v \in [0,1]$. Let T be as defined in (1). Then for any adaptive strategy* ADAP *which uses budget at most γ^i, we have*

$$\sum_{v\in\Pi_i\setminus\Pi_i(\mathsf{ADAP})} \mathbb{E}[X_v] \;\ge\; (C-1)(T-\epsilon)-\epsilon.$$

Proof of Lemma 9. Since ADAP uses budget at most γ^i, the set $\Pi_i(\mathsf{ADAP}) \subseteq \Pi_i$ can always be visited by a tour of length at most γ^i. Consider the first tour π_1 found by Algorithm 2. Since the tour with length at most γ^i that visits the set $\Pi_i(\mathsf{ADAP})$ is a valid Orienteering tour when π_1 is found, it follows from Lemma 6 that $\sum_{v\in\Pi_i(\mathsf{ADAP})} \mathbb{E}[X_v] \le \epsilon + \sum_{v\in\pi_1} \mathbb{E}[X_v]$. For any $s \in \{2,3,\cdots,C\}$, since π is an Orienteering tour in $V\setminus\Pi_i$ of length at most γ^i with $\sum_{v\in\pi} \mathbb{E}[X_v] = T$, Lemma 6 implies that $\sum_{v\in\pi_s} \mathbb{E}[X_v] \ge T-\epsilon$. Therefore, a simple calculation gives

$$\sum_{v\in\Pi_i\setminus\Pi_i(\mathsf{ADAP})} \mathbb{E}[X_v] \;=\; \sum_{v\in\Pi_i} \mathbb{E}[X_v] - \sum_{v\in\Pi_i(\mathsf{ADAP})} \mathbb{E}[X_v] \;\ge\; (C-1)(T-\epsilon)-\epsilon,$$

which finishes the proof of Lemma 9. ◀

4 Stoch-Reward k-TSP

In this section we prove Theorem 1, which is restated below for convenience.

▶ **Theorem 1.** *The Stoch-Reward k-TSP problem has a non-adaptive $O(1)$-approximation algorithm.*

To prove this theorem we carefully choose the parameters of our Meta-Algorithm from last section.

4.1 The Algorithm

Algorithm 3 ALG$_{\text{Stoch-Reward}}$ for Stoch-Reward k-TSP problem.

```
1  Pre-processing stage:
2  set $\gamma \leftarrow 1.1, \epsilon \leftarrow 1/10^5, \Pi \leftarrow \emptyset$, $\ell = \lfloor \log k \rfloor$, and $C \leftarrow 6000$ ;
3  for phase $i = 0, 1, \cdots$ do
4      set $\Pi_{i,-1} \leftarrow \emptyset$ ;
5      for scale $j = 0, \cdots, \ell$ do
6          set profit $w_v^j = \mathbb{E}\left[\min\left\{R_v \cdot 2^j/k, 1\right\}\right] \cdot \mathbf{1}[v \in V \setminus \Pi]$ ;    /* Scale $j$ truncates
             at $k/2^j$ */
7          set $\Pi_{i,j} \leftarrow \emptyset$ ;
8          for repetition $s = 1, 2, \cdots, C$        /* Constant repetitions of ALG$_{\text{Bicrit-Orient}}$ */
9           do
10             use ALG$_{\text{Bicrit-Orient}}$ to find tour $\pi_{i,j,s}$ with budget $\gamma^i$, profit $\{w_v^j\}_{v \in V}$, and
                error $\epsilon$ ;
11             append tour $\pi_{i,j,s}$ to $\Pi_{i,j}$, i.e., $\Pi_{i,j} \leftarrow \Pi_{i,j} \circ \pi_{i,j,s}$ ;
12             reset $w_v^j = 0$ for $v \in \Pi_{i,j}$ ;
13         end
14          /* Check whether $j$ is a "critical" scale */
15         use ALG$_{\text{Orient}}$ to find tour $\pi_{i,j}^{\text{Ori}}$ with budget $\gamma^i$ and profit $\{w_v^j\}_{v \in V}$ ;
16         set $T_{i,j} \leftarrow \sum_{v \in \pi_{i,j}^{\text{Ori}}} w_v^j$ ;
17         if $T_{i,j} \geq 1/300$ or $j = \ell$ then
18             append tour $\Pi_i := \Pi_{i,j} \cup \Pi_{i,j-1}$ to $\Pi$, i.e., $\Pi \leftarrow \Pi \circ \Pi_i$ ;
19             Break ;
20         end
21     end
22 end
23
24 Probing stage:
25 for phase $i = 0, 1, \cdots$ do
26     visit vertices in the order of $\Pi_i$ and apply the following selection and stopping
         criteria ;
27     Selection Criterion: select every vertex visited ;
28     Stopping Criterion: total reward reaches $k$ ;
29 end
30 Return set of vertices selected
```

We assume without loss of generality that the stochastic reward $R_v \leq k$ almost surely for each vertex $v \in V$. Our algorithm ALG$_{\text{Stoch-Reward}}$ for Stoch-Reward k-TSP problem is given in Algorithm 3. It is an instantiation of the Meta-Algorithm ALG$_{\text{Meta}}$ (Algorithm 1) by setting the phase parameter $\gamma = 1.1$, number of scales $\ell = \lfloor \log k \rfloor$, and number of repetitions $C = 6000$. For each scale j in phase i, we set the random variable X_v^j for any vertex $v \in V \setminus \Pi$ in ALG$_{\text{Meta}}$ to be the stochastic reward R_v truncated at $k/2^j$ and then scaled down to $[0, 1]$. We identify the "critical" scale j_{crit} in ALG$_{\text{Meta}}$ as follows: For each scale j in phase i, denote $\Pi_{i,j}$ the $C = 6000$ repetitions of ALG$_{\text{Bicrit-Orient}}$ and let $T_{i,j}$ be the profit obtained by the 3-approximation Orienteering algorithm ALG$_{\text{Orient}}$ in $V \setminus (\Pi \cup \Pi_{i,j})$. The "critical" scale j_{crit} is the smallest scale j such that $T_{i,j} \geq 1/300$, i.e., sufficient profit remains outside even after C repetitions. We add the two tours corresponding to the "rich" scale j_{crit} and the "poor" scale $j_{\mathsf{crit}} - 1$ into Π. In the case when there is no scale j with $T_{i,j} \geq 1/300$, we simply set j_{crit} to be the last scale ℓ. In the probing stage, the selection and the stopping criteria are straightforward: we collect reward from every visited vertex and stop when the total reward reaches k.

4.2 Proof of Theorem 1

Recall from Section 3.1 that to prove Theorem 1, we only need to prove the precondition in Lemma 7. The remainder of this section proves this precondition for $\mathsf{ALG}_{\mathsf{Stoch\text{-}Reward}}$ as given in the following Lemma 10.

▶ **Lemma 10.** *For $\gamma = 1.1$, any phase $i > 0$ in the probing stage of* $\mathsf{ALG}_{\mathsf{Stoch\text{-}Reward}}$ *satisfies*

$$u_i(\sigma_{i-1}) \leq 100 \cdot u_i^*(\sigma_{i-1}) + \frac{u_{i-1}(\sigma_{i-1})}{\gamma^2}. \tag{2}$$

Before proving Lemma 10, we discuss the high-level intuition of our proof.

Intuition. Assume without loss of generality that $u_i^*(\sigma_{i-1}) < 0.01$, i.e., OPT finds k reward within budget γ^i with probability at least 0.99, as otherwise the lemma trivially holds. Now the plan is to show that with constant probability, $\mathsf{ALG}_{\mathsf{Stoch\text{-}Reward}}$ finds more reward than OPT restricted to budget γ^i, even when all rewards in σ_{i-1} are given to OPT for free. This allows us to focus on the remaining graph with vertex set $V_i := V \setminus \sigma_{i-1}$ where we repeat $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ for different scales.

Our arguments rely on the notion of "richness". We call a scale j "rich" if $T_{i,j} \geq 1/300$ and otherwise "poor". A scale being poor indicates that not much reward can be collected outside the C repetitions of $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ for that scale, in which case we can use Lemma 8 to argue that OPT cannot find much reward outside. A scale being rich implies that each repetition of $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ for that scale finds a significant amount of reward, in which case we can apply Lemma 9 to argue that $\mathsf{ALG}_{\mathsf{Stoch\text{-}Reward}}$ finds much more reward than OPT in the C repetitions for that scale. Our critical scale j_{crit} corresponds to the transition from poor to rich scales. Since the algorithm includes both j_{crit} and $j_{\mathsf{crit}} - 1$, roughly the reason why our analysis works is that we use the poor scale $j_{\mathsf{crit}} - 1$ to argue OPT cannot find much reward outside our tours and we use the rich scale j_{crit} to argue OPT cannot find much more reward inside. The final analysis has to do some case analysis depending on whether the transition ever happens or not.

Proof of Lemma 10. We fix any outcome σ_{i-1} of vertices visited by $\mathsf{ALG}_{\mathsf{Stoch\text{-}Reward}}$ in the first $i-1$ phases in its probing stage. The lemma trivially holds in the case where $u_i^*(\sigma_{i-1}) \geq 0.01$ as we have $100u_i^*(\sigma_{i-1}) \geq 1$. If $u_{i-1}(\sigma_{i-1}) = 0$ which means that $\mathsf{ALG}_{\mathsf{Stoch\text{-}Reward}}$ already collects reward k before entering phase i in the probing stage, then $u_i(\sigma_{i-1}) = 0$ and again the lemma trivially holds. We therefore assume that $u_i^*(\sigma_{i-1}) < 0.01$ and that $u_{i-1}(\sigma_{i-1}) = 1$. Now proving Lemma 10 is equivalent to proving

$$u_i(\sigma_{i-1}) \leq 100u_i^*(\sigma_{i-1}) + 1/\gamma^2. \tag{3}$$

To prove (3), we need the following notation. Denote $V_i := V \setminus \sigma_{i-1}$ the vertex set of the remaining graph where vertices in σ_{i-1} are excluded. Denote $\overline{\Pi}_i(\mathsf{OPT}) \subseteq V_i \setminus \Pi_i$ the (random) set of vertices visited by OPT outside $\sigma_{i-1} \cup \Pi_i$ within budget γ^i, and denote $\Pi_i(\mathsf{OPT}) \subseteq \Pi_i$ the (random) set of vertices visited by OPT inside Π_i. We consider three cases:

Case (1): (Scale 0 is rich) $T_{i,0} \geq 1/300$. In this case, our algorithm appends tour $\Pi_i := \Pi_{i,0}$ to Π (recall that $\Pi_{i,-1} = \emptyset$), and these will be the phase i vertices visited in the probing stage. We show that $T_{i,0} \geq 1/300$ implies that each repetition of $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ has large expected reward (notice the random rewards are not truncated at scale 0). As we repeat $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ for $C = 6000$ times, the tour $\Pi_{i,0}$ has expected reward much larger than the target k.
Since $T_{i,0}$ is the profit of a valid Orienteering tour with length at most γ^i, for each $s \in \{1, \ldots, C\}$ we have $T_{i,0,s} \geq T_{i,0} - \epsilon \geq 1/300 - \epsilon$, where $\epsilon = 1/10^5$ is the small error term for $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ in Lemma 6. Thus,

$$\sum_{v \in \Pi_{i,0}} \mathbb{E}\left[\min\left\{R_v/k, 1\right\}\right] \quad \geq \quad 20 - 6000\epsilon \quad \geq \quad 19.$$

Notice that $R_v/k \in [0,1]$, so applying Chernoff bound (Theorem 5) we have

$$1 - u_i(\sigma_{i-1}) \quad \geq \quad \mathbb{P}\Big[\sum_{v \in \Pi_{i,0}} R_v \geq k\Big] \quad = \quad \mathbb{P}\Big[\sum_{v \in \Pi_{i,0}} \min\left\{R_v/k, 1\right\} \quad \geq \quad 1\Big] \quad \geq \quad 0.9,$$

which means $u_i(\sigma_{i-1}) \leq 0.1 \leq 1/\gamma^2$. This proves (3) and finishes the proof of Lemma 10 in this case.

Case (2): (Scale ℓ is poor) $T_{i,j} \leq 1/300$ for every scale $j = 0, \cdots, \ell$. In this case, our algorithm adds $\Pi_i := \Pi_{i,\ell} \cup \Pi_{i,\ell-1}$ to Π, and these will be the phase i vertices visited in the probing stage. We argue that the assumption of $T_{i,\ell} \leq 1/300$ implies that with constant probability OPT finds *no* reward outside $\sigma_{i-1} \cup \Pi_i$ within budget γ^i. If OPT still manages to find k reward, then all k reward must come from vertices in $\sigma_{i-1} \cup \Pi_i$, in which case $\mathsf{ALG}_{\mathsf{Stoch\text{-}Reward}}$ also finds k reward. In this case our argument already works with the vertices $\Pi_{i,\ell}$ added to Π, i.e., we do not even need vertices in $\Pi_{i,\ell-1}$.
Since $\mathsf{ALG}_{\mathsf{Orient}}$ is a 3-approximation Orienteering algorithm, for any set of vertices $S \subseteq V_i \setminus \Pi_i$ that can be visited within budget γ^i, we have $\sum_{v \in S} \mathbb{E}\left[\min\left\{R_v \cdot 2^\ell/k, 1\right\}\right] \leq 3T_{i,\ell} \leq 0.01$. Since $\ell = \lfloor \log k \rfloor$, we have $k/2^\ell \in [1,2]$, and therefore

$$\sum_{v \in S} \mathbb{E}[\min\{R_v, 1\}] \quad \leq \quad \sum_{v \in S} \mathbb{E}\left[\min\left\{2 \cdot R_v \cdot 2^\ell/k, 1\right\}\right] \quad \leq \quad 0.02. \tag{4}$$

Recall that $\overline{\Pi}_i(\mathsf{OPT}) \subseteq V_i \setminus \Pi_i$ denotes the (random) set of vertices visited by OPT outside $\sigma_{i-1} \cup \Pi_i$ within budget γ^i. Since each $\min\{R_v, 1\} \in \{0, 1\}$, the best probability of obtaining truncated reward at least 1 in $V_i \setminus \Pi_i$ within budget γ^i is achieved by a non-adaptive strategy. Therefore, Markov's inequality together with (4) implies that

$$\mathbb{P}\Big[\sum_{v \in \overline{\Pi}_i(\mathsf{OPT})} \min\{R_v, 1\} \geq 1\Big] \ \leq \ 0.02.$$

Since $u_i^*(\sigma_{i-1}) < 0.01$, we have that with probability at least $1 - 0.01 - 0.02 = 0.97$, OPT finds k reward in V but 0 reward outside $\sigma_{i-1} \cup \Pi_i$. In this case, $\mathsf{ALG}_{\mathsf{Stoch\text{-}Reward}}$ also finds k reward among vertices visited in the first i phases. So we have $u_i(\sigma_{i-1}) \leq 1 - 0.97 = 0.03 \leq 1/\gamma^2$, which establishes (3) in this case.

Case (3): (Transition from poor to rich scale at j_{crit}) $T_{i,0} \leq 1/300$ but $T_{i,j} > 1/300$ for some $j \in [\ell]$. In this case, let $j_{\mathsf{crit}} = \min\left\{j \in [\ell] : T_{i,j} > 1/300\right\}$ be our critical scale. The algorithm appends $\Pi_i := \Pi_{i,j_{\mathsf{crit}}-1} \cup \Pi_{i,j_{\mathsf{crit}}}$ to Π. To prove that $\mathsf{ALG}_{\mathsf{Stoch\text{-}Reward}}$ will not continue to phase $i+1$ with constant probability, we show that the following two events happen with constant probability:

1. OPT doesn't find too much reward outside $\sigma_{i-1} \cup \Pi_i$ within budget γ^i.
2. $\mathsf{ALG}_{\mathsf{Stoch\text{-}Reward}}$ finds much more reward inside Π_i than OPT does since it is restricted to budget γ^i.

We show that the first event follows from $T_{i,j_{\mathsf{crit}}-1} \le 1/300$ while the second event follows from $T_{i,j_{\mathsf{crit}}} \ge 1/300$. From these we conclude that with constant probability, $\mathsf{ALG}_{\mathsf{Stoch\text{-}Reward}}$ obtains at least as much reward as OPT restricted to budget γ^i.

We first argue that OPT doesn't find too much reward outside $\sigma_{i-1} \cup \Pi_i$. Specifically, we prove that

$$\mathbb{P}\Big[\sum_{v \in \overline{\Pi}_i(\mathsf{OPT})} R_v < k/2^{j_{\mathsf{crit}}-1}\Big] \;\ge\; 0.5. \tag{5}$$

Notice that $T_{i,j_{\mathsf{crit}}-1} \le 1/300$ together with the fact that $\mathsf{ALG}_{\mathsf{Orient}}$ is a 3-approximation for Orienteering implies that for any set of vertices $S \subseteq V_i \setminus \Pi_i$ that can be visited within distance γ^i, we have

$$\sum_{v \in S} \mathbb{E}\left[\min\left\{R_v \cdot 2^{j_{\mathsf{crit}}-1}/k, 1\right\}\right] \le 0.01.$$

Since the random variables $\min\left\{R_v \cdot 2^{j_{\mathsf{crit}}-1}/k, 1\right\} \in [0,1]$, applying Lemma 8 we have

$$\mathbb{P}\Big[\sum_{v \in \overline{\Pi}_i(\mathsf{OPT})} \min\left\{R_v \cdot 2^{j_{\mathsf{crit}}-1}/k, 1\right\} \ge 1\Big] \quad \le \quad 2\exp\left(-\frac{0.99^2/2}{0.01 + 0.99/3}\right) \quad \le \quad 0.5,$$

which immediately implies (5).

Now we argue that inside Π_i, we find much more reward than OPT does when it's restricted to budget γ^i. Specifically, we prove that

$$\mathbb{P}\Big[\sum_{v \in \Pi_i \setminus \Pi_i(\mathsf{OPT})} R_v \ge k/2^{j_{\mathsf{crit}}-1}\Big] \;\ge\; 0.8, \tag{6}$$

where recall that $\Pi_i(\mathsf{OPT}) \subseteq \Pi_i$ is the (random) set of vertices visited by OPT inside Π_i within budget γ^i. Notice that $T_{i,j_{\mathsf{crit}}} > 1/300$ together with Lemma 9 implies that

$$\sum_{v \in \Pi_i \setminus \Pi_i(\mathsf{OPT})} \mathbb{E}[\min\{R_v \cdot 2^{j_{\mathsf{crit}}}/k, 1\}] \quad \ge \quad (6000-1)\cdot(T_{i,j_{\mathsf{crit}}} - \epsilon) - \epsilon \quad \ge \quad 19.$$

Applying Chernoff bound (Theorem 5), we have

$$\mathbb{P}\Big[\sum_{v \in \Pi_i \setminus \Pi_i(\mathsf{OPT})} \min\{R_v \cdot 2^{j_{\mathsf{crit}}}/k, 1\} \ge 2\Big] \;\ge\; 0.8,$$

which implies (6).

Now we complete the proof of (3) in this final case. From (5) and (6) and our assumption that $u_i^*(\sigma_{i-1}) < 0.01$, we have that with probability at least $1 - 0.5 - 0.2 - 0.01 \ge 1/4$, all the following three events hold: (1) $\sum_{v \in \overline{\Pi}_i(\mathsf{OPT})} R_v < k/2^{j_{\mathsf{crit}}-1}$, (2) $\sum_{v \in \Pi_i \setminus \Pi_i(\mathsf{OPT})} R_v \ge k/2^{j_{\mathsf{crit}}-1}$, and (3) OPT obtains at least k reward within budget γ^i. When all these three events hold, $\mathsf{ALG}_{\mathsf{Stoch\text{-}Reward}}$ also finds at least k reward before visiting any vertex from phase $i+1$. Therefore, $u_i(\sigma_{i-1}) \le 3/4 \le 1/\gamma^2$, and this completes the proof of Lemma 10. ◀

5 Stoch-Cost k-TSP

In this section we prove Theorem 2, which is restated below for convenience. Throughout this section, we will remove the restriction that a vertex v can only be selected if we are currently at v because this is equivalent to the original problem up to a factor of 2.

▶ **Theorem 2.** *The Stoch-Cost k-TSP problem has a non-adaptive $O(1)$-approximation algorithm.*

Recall, an additional challenge for Stoch-Cost k-TSP is that there is no obvious way to truncate the cost distributions C_v, even if the remaining target k' is known. Truncating at the "average" cost per remaining reward (i.e., $\mathbb{P}[C_v \leq O(\gamma^i/k')]$) will fail when some vertices in the optimal tour have costs much smaller than $O(\gamma^i/k')$, while the other vertices have much higher costs. We overcome this by considering $\mathbb{P}[C_v \leq O(\gamma^i/2^j)]$ for all possible scales $j \in \{0, \cdots, i \cdot \log \gamma + \log n\}$. To identify a "critical" scale, we evaluate the maximum target a tour at a given scale can get with constant probability within cost budget 2^i. We show this can be approximately computed via dynamic programming.

The rest of this section is devoted to proving Theorem 2.

5.1 The Algorithm

Our algorithm $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$ for Stoch-Cost k-TSP is given in Algorithm 4. $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$ is an instantiation of our Meta-Algorithm $\mathsf{ALG}_{\mathsf{Meta}}$ in Algorithm 1 by setting the phase parameter $\gamma = 1.1$ and number of repetitions $C = 6000$. The number of scales in phase $i \geq 0$ will be $\ell_i = \lfloor i \cdot \log \gamma + \log n \rfloor$. (Notice, unlike Stoch-Reward k-TSP, the number of scales changes with phase.) For scale j in phase i, we set a random variable X_v^j for vertex $v \in V \setminus \Pi$ in $\mathsf{ALG}_{\mathsf{Meta}}$ to be the indicator variable that cost $C_v \leq \gamma^i/2^j$.

We identify a "critical" scale $\widetilde{j}_{\mathsf{crit}}$ as follows: For any phase $i \geq 0$ and scale $j \in \{0, \cdots, \ell_i\}$, define $Y_{i,j} \in \mathbb{Z}_{\geq 0}$ to be the maximum number of vertices that can be selected from $\Pi_{i,j} \cup \Pi_{i,j-1}$ with probability at least 0.2 within cost budget $3\gamma^i$. Ideally, we want to compute $Y_{i,j}$ for every scale $j \in \{0, \cdots, \ell_i\}$ and set the critical scale to be the one that maximizes $Y_{i,j}$. Unfortunately, $Y_{i,j}$ cannot be computed efficiently as the corresponding problem is NP-Hard. To get around this issue, we compute an approximate value $\widetilde{Y}_{i,j}$ in Step 15 of $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$ via a dynamic programming sub-procedure $\mathsf{ALG}_{\mathsf{DP}}$. We discuss the details of $\mathsf{ALG}_{\mathsf{DP}}$ in Section 5.3, where we prove the following Lemma 11 which roughly says that the $\widetilde{Y}_{i,j}$ computed by $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$ is a reasonably good approximation of $Y_{i,j}$.

▶ **Lemma 11.** *For any phase $i \geq 0$ and any scale $j \in \{0, \cdots, \ell_i\}$, the approximate value $\widetilde{Y}_{i,j}$ computed in Step 15 of $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$ satisfies that (1) $\widetilde{Y}_{i,j} \geq Y_{i,j}$, and (2) $\widetilde{Y}_{i,j}$ vertices can be selected from $\Pi_{i,j} \cup \Pi_{i,j-1}$ within cost budget $6\gamma^i$ with probability at least 0.2.*

After computing $\widetilde{Y}_{i,j}$ for each scale $j \in \{0, \cdots, \ell_i\}$, we simply set $\widetilde{j}_{\mathsf{crit}}$ to be the scale that maximizes $\widetilde{Y}_{i,j}$ and add the two tours corresponding to scales $\widetilde{j}_{\mathsf{crit}} - 1$ and $\widetilde{j}_{\mathsf{crit}}$ into Π.

In the probing stage, the Stopping Criterion is natural: we stop whenever the number of selected vertices reaches k. The selection process needs some care since not all vertices visited in the previous phases are selected. Our algorithm therefore runs two selection processes consecutively: In Selection-Process 1, we select as many unselected vertices from those visited in the previous phases within total cost γ^i. This is to ensure that we select from σ_{i-1} at least as many vertices as OPT restricted to budget γ^i. In Selection-Process 2, we select as many vertices as possible from Π_i within total cost $6\gamma^i$. The above Lemma 11 guarantees that at least $\widetilde{Y}_{i,\widetilde{j}_{\mathsf{crit}}}$ vertices can be selected in this process with probability at least 0.2.

Algorithm 4 $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$ for Stoch-Cost k-TSP problem.

```
Pre-processing stage:
set $\gamma \leftarrow 1.1$, $\epsilon \leftarrow 1/10^5$, $\Pi \leftarrow \emptyset$ and $C \leftarrow 6000$ ;
for phase $i = 0, 1, \cdots$ do
    set $\Pi_{i,-1} \leftarrow \emptyset$ ;
    for scale $j = 0, \cdots, \ell_i$, where $\ell_i = \lfloor i \cdot \log \gamma + \log n \rfloor$ do
        set profit $w_v^j = \mathbb{P}[C_v \leq \gamma^i / 2^j] \cdot \mathbb{1}[v \in V \setminus \Pi]$ ;    /* Scale $j$ "truncates" at $\gamma^i/2^j$ */
        set $\Pi_{i,j} \leftarrow \emptyset$ ;
        for repetition $s = 1, 2, \cdots, C$    /* Constant repetitions of $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ */
         do
            use $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ to find tour $\pi_{i,j,s}$ with budget $\gamma^i$, profit $\{w_v^j\}_{v \in V}$ and error $\epsilon$ ;
            append tour $\pi_{i,j,s}$ to $\Pi_{i,j}$, i.e., $\Pi_{i,j} \leftarrow \Pi_{i,j} \circ \pi_{i,j,s}$ ;
            reset $w_v^j = 0$ for $v \in \Pi_{i,j}$ ;
        end
        /* Approximately compute the maximum number of vertices that can be selected from $\Pi_{i,j} \cup \Pi_{i,j-1}$ within cost budget $3\gamma^i$ and with probability at least 0.2 */
        find the largest integer $\widetilde{Y}_{i,j} \leq n$ such that $\mathsf{ALG}_{\mathsf{DP}}(\widetilde{Y}_{i,j}, 3\gamma^i, \Pi_{i,j} \cup \Pi_{i,j-1}) \geq 0.2$ ;
    end
    /* Identify a "critical" scale */
    set $\widetilde{j}_{\mathsf{crit}} \leftarrow \arg\max_j \widetilde{Y}_{i,j}$ and $\Pi_i \leftarrow \Pi_{i,\widetilde{j}_{\mathsf{crit}}} \cup \Pi_{i,\widetilde{j}_{\mathsf{crit}}-1}$ ;
    append tour $\Pi_i$ to $\Pi$, i.e., $\Pi \leftarrow \Pi \circ \Pi_i$ ;
end

Probing stage:
for phase $i = 0, 1, \cdots$ do
    set $\sigma_{i-1} \leftarrow \bigcup_{t=0}^{i-1} \Pi_t$ ;
    visit vertices in the order of $\Pi_i$ and apply the following selection and stopping criteria ;
    /* Select (unselected) vertices visited in previous phases */
    Selection-Process 1: select as many vertices as possible from $\sigma_{i-1}$ within total cost $\gamma^i$ ;
    /* Select vertices visited in the current phase */
    Selection-Process 2: select as many vertices as possible from $\Pi_i$ within total cost $6\gamma^i$ ;
    Stopping Criterion: total number of vertices selected reaches $k$ ;
end
Return the set of selected vertices
```

5.2 Proof of Theorem 2

Recall from Section 3.1 that to prove Theorem 2, we only need to prove the precondition in Lemma 7. The remainder of this section proves this precondition for $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$ as stated in the following Lemma 12.

▶ **Lemma 12.** *For $\gamma = 1.1$, any phase $i > 0$ in the probing stage of Stoch-Cost k-TSP satisfies*

$$u_i(\sigma_{i-1}) \leq 100u_i^*(\sigma_{i-1}) + \frac{u_{i-1}(\sigma_{i-1})}{\gamma^2}. \tag{7}$$

Before proving Lemma 12, we need some notation.

Notation. For any (possibly adaptive) algorithm ADAP, we say ADAP has a *distance budget* of B if it is allowed to travel a total distance of at most B; we say ADAP has a *cost budget* of B if it is allowed to select vertices up to a total cost of B; we say ADAP has a *total budget* of B if its total distance travelled *plus* the total cost of selecting vertices is restricted to be at most B. An algorithm satisfying a budget constraint is said to be *within* that budget. For any phase i, denote $V_i := V \setminus \sigma_{i-1}$ the set of vertices in the remaining graph where vertices in σ_{i-1} are excluded. For any fixed outcome σ_{i-1} and any target $Y \in \mathbb{Z}_{\geq 0}$, denote $p^*_{i,Y}(\sigma_{i-1})$ the probability that OPT selects at least Y vertices from V_i within total budget γ^i and $p_{i,Y}(\sigma_{i-1})$ the probability that the tour $\Pi_i \subseteq V_i$ found by $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$ contains Y vertices which can be selected within cost budget $6\gamma^i$.

The proof of Lemma 12 relies on the following Lemma 13, which says that if OPT selects Y vertices in V_i within total budget γ^i with probability at least 0.9, then we can select Y vertices in Π_i within cost budget $6\gamma^i$ with probability at least 0.2. The proof of Lemma 12 from Lemma 13 is standard, see Appendix C.

▶ **Lemma 13.** *For any phase $i \geq 0$, any target $Y \in \mathbb{Z}_{\geq 0}$ and any outcome σ_{i-1} of vertices visited in the previous $i-1$ phases, if $p^*_{i,Y}(\sigma_{i-1}) \geq 0.9$ then we have $p_{i,Y}(\sigma_{i-1}) \geq 0.2$.*

We need some notation to prove Lemma 13.

Notation. We say a vertex $v \in V$ is *qualified* for scale $j \in \{0, \cdots, \ell_i\}$ if its cost $C_v \leq \gamma^i/2^j$. For each scale $j \in \{0, \cdots, \ell_i\}$, denote $\pi^*_{i,j}$ the optimal $\mathsf{Orienteering}$ tour in $V_i \backslash \Pi_{i,j}$ with budget γ^i where each vertex $v \in V_i \setminus \Pi_{i,j}$ has profit $\mathbb{P}[C_v \leq \gamma^i/2^j]$. Define $T^*_{i,j} := \sum_{v \in \pi^*_{i,j}} \mathbb{P}[C_v \leq \gamma^i/2^j]$ and $T_{i,j} := \sum_{v \in \Pi_{i,j}} \mathbb{P}[C_v \leq \gamma^i/2^j]$ to be the total $\mathsf{Orienteering}$ profit of tour $\pi^*_{i,j}$ and $\Pi_{i,j}$, respectively. Denote $\overline{\Pi}_{i,j}(\mathsf{OPT}) \subseteq V_i \setminus (\Pi_{i,j} \cup \Pi_{i,j-1})$ the (random) set of vertices visited by OPT outside $\sigma_{i-1} \cup (\Pi_{i,j} \cup \Pi_{i,j-1})$ within total budget γ^i, and $\Pi_{i,j}(\mathsf{OPT}) \subseteq \Pi_{i,j} \cup \Pi_{i,j-1}$ the (random) set of vertices visited by OPT inside $\Pi_{i,j} \cup \Pi_{i,j-1}$ within total budget γ^i.

Intuition. We discuss our high-level proof strategy for Lemma 13. Our arguments again rely on the notion of "richness". Recall from above that $T_{i,j}$ denotes the $\mathsf{Orienteering}$ profit of tour $\Pi_{i,j}$ at scale j. We call a scale j "rich" if $T_{i,j} \geq Y$ and otherwise "poor". A scale j being poor roughly (but not quite) indicates that OPT cannot find enough low-cost vertices qualified for scale j outside the tour $\Pi_{i,j}$. A scale j being rich implies that $\Pi_{i,j}$ contains enough vertices that are qualified for scale j, which is an immediate consequence of Jogdeo-Samuels inequality (Theorem 3). We plan to find a critical scale j_{crit} that corresponds to the transition from rich to poor scales. Notice that such a critical scale j_{crit} might be different from the critical scale $\widetilde{j}_{\mathsf{crit}}$ found by our algorithm, but we show that it suffices to argue about j_{crit} since $\widetilde{j}_{\mathsf{crit}}$ is only better.

To argue about j_{crit}, we consider the tours corresponding to both scales j_{crit} and $j_{\mathsf{crit}} - 1$. Roughly we use the poor scale j_{crit} to argue that OPT cannot find enough low-cost vertices outside these tours and we use the rich scale $j_{\mathsf{crit}}-1$ to argue that we have enough replacements for these low-cost vertices without paying too much cost. Our final analysis is a case analysis depending on whether the transition ever happens or not. The proof here is more involved than that in Section 4 as we also need to take into account the amount of $\mathsf{Orienteering}$ profit outside the $C = 6000$ repetitions of $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ for each scale j.

Proof of Lemma 13. Recall that for any phase $i \geq 0$ and scale $j \in \{0, \cdots, \ell_i\}$, we defined $Y_{i,j}$ to be the maximum number of vertices that can be selected from $\Pi_{i,j} \cup \Pi_{i,j-1}$ with probability at least 0.2 within cost budget $3\gamma^i$. We show in the following that $p^*_{i,Y}(\sigma_{i-1}) \geq 0.9$ implies there exists a "critical" scale $j_{\mathsf{crit}} \in \{0, \cdots, \ell_i\}$ with $Y_{i,j_{\mathsf{crit}}} \geq Y$. This critical scale j_{crit} might be different from the critical scale $\widetilde{j}_{\mathsf{crit}}$ identified by $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$. But since $\widetilde{j}_{\mathsf{crit}}$ maximizes $\widetilde{Y}_{i,j}$ among all scales $j \in \{0, \cdots, \ell_i\}$, it follows from property (1) in Lemma 11 that $\widetilde{Y}_{i,\widetilde{j}_{\mathsf{crit}}} \geq \widetilde{Y}_{i,j_{\mathsf{crit}}} \geq Y_{i,j_{\mathsf{crit}}} \geq Y$. Now using property (2) in Lemma 11 we have that $\widetilde{Y}_{i,\widetilde{j}_{\mathsf{crit}}} \geq Y$ vertices can be selected from $\Pi_{i,\widetilde{j}_{\mathsf{crit}}} \cup \Pi_{i,\widetilde{j}_{\mathsf{crit}}-1}$ within cost budget $6\gamma^i$ with probability at least 0.2. It follows that $p_{i,Y}(\sigma_{i-1}) \geq 0.2$. Therefore, the existence of a critical scale j_{crit} with $Y_{i,j_{\mathsf{crit}}} \geq Y$ would imply Lemma 13.

In the following, we consider three different cases and prove the existence of such a critical scale j_{crit} with $Y_{i,j_{\mathsf{crit}}} \geq Y$ in each case.

Case (1): (Scale 0 is poor) $T_{i,0} < Y$. We show in this case that $p^*_{i,Y}(\sigma_{i-1}) \geq 0.9$ implies that scale 0 is a "critical" scale with $Y_{i,0} \geq Y$. Notice that any vertex $v \in V_i$ not qualified for scale 0 has cost $C_v > \gamma^i$. Therefore, OPT cannot select any vertex that is not qualified for scale 0 within total budget γ^i.

We start by showing that $T^*_{i,0} \leq 0.1$. To prove this, we assume for the purpose of contradiction that $T^*_{i,0} > 0.1$. It follow from Lemma 8 that

$$\mathbb{P}\Big[\,\big|\big\{v \in \overline{\Pi}_{i,0}(\mathsf{OPT}) : C_v \leq \gamma^i\big\}\big| \leq 200T^*_{i,0}\Big] \;\geq\; 0.9. \tag{8}$$

From Lemma 9 we have that

$$\sum_{v \in \Pi_{i,0} \setminus \Pi_{i,0}(\mathsf{OPT})} \mathbb{P}[C_v \leq \gamma^i] \quad\geq\quad (6000-1)\cdot(T^*_{i,0} - \epsilon) - \epsilon \quad\geq\quad 5000T^*_{i,0}.$$

So it follows from Chernoff bound (Theorem 5) that

$$\mathbb{P}\Big[\,\big|\big\{v \in \Pi_{i,0} \setminus \Pi_{i,0}(\mathsf{OPT}) : C_v \leq \gamma^i\big\}\big| \geq 500T^*_{i,0}\Big] \;\geq\; 0.9. \tag{9}$$

Since $T_{i,0} < Y$, from Theorem 3 we have that

$$\mathbb{P}\Big[\,\big|\big\{v \in \Pi_{i,0} : C_v \leq \gamma^i\big\}\big| \leq Y\Big] \;\geq\; 0.5. \tag{10}$$

Now we count the number of vertices found by OPT that are qualified for scale 0 within total budget γ^i. It follows from union bound that with probability at least 0.2, all three events in (8), (9) and (10) hold, in which case OPT finds at most $Y - 300T^*_{i,0}$ vertices qualified for scale 0 within total budget γ^i. Therefore, in order to select at least Y vertices, OPT needs to select vertices that are not qualified for scale 0 within total budget γ^i, which is a contradiction to the assumption that $p^*_{i,Y}(\sigma_{i-1}) \geq 0.9$.

Therefore we must have $T^*_{i,0} \leq 0.1$. Applying Markov's inequality, the probability that OPT finds any vertex qualified for scale 0 in $V_i \setminus \Pi_{i,0}$ is upper bounded by $T^*_{i,0} \leq 0.1$. When this happens and when OPT selects Y vertices within total budget γ^i, all vertices selected by OPT are from $\Pi_{i,0}$. Therefore, with probability at least $p^*_{i,Y}(\sigma_{i-1}) - 0.1 \geq 0.8$, we can select Y vertices from $\Pi_{i,0}$ within cost budget γ^i which implies that $Y_{i,0} \geq Y$.

Case (2): (All scales are rich) $T_{i,j} \geq Y$ for every scale $j = 0, \cdots, \ell_i$. In this case we show that ℓ_i is a "critical" scale with $Y_{i,\ell_i} \geq Y$. Notice that selecting any Y vertices qualified for scale ℓ_i has cost at most $Y \cdot \gamma^i / 2^{\ell_i} \leq 2 \leq 6\gamma^i$. Since $T_{i,\ell_i} \geq Y$, it follows from Theorem 3 that with probability no less than 0.5, at least Y vertices in Π_{i,ℓ_i} are qualified for scale ℓ_i (notice we don't even need the tour $\Pi_{i,\ell_i - 1}$ in this case). This implies that $Y_{i,\ell_i} \geq Y$.

Case (3): (Transition from rich to poor scale at j_{crit}) $T_{i,0} \geq Y$ but $T_{i,j} < Y$ for some $j \in [\ell_i]$. In this case, let $j_{\mathsf{crit}} = \arg\min_j \{j \in [\ell_i] : T_{i,j} < Y\}$. We show in the following that j_{crit} is a "critical" scale with $Y_{i,j_{\mathsf{crit}}} \geq Y$. To prove this, we show that the following two events happen with constant probability:

1. $T_{i,j_{\mathsf{crit}}} < Y$ implies that OPT doesn't find enough vertices *qualified* for scale j_{crit}. This gives a lower bound on the cost of the set of vertices selected by OPT within total budget γ^i.
2. $T_{i,j_{\mathsf{crit}}-1} \geq Y$ implies that $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$ finds enough vertices *qualified* for scale $j_{\mathsf{crit}} - 1$. This can be used to upper bound the cost of $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$.

We first consider the sub-case where $T^*_{i,j_{\mathsf{crit}}} \leq 0.1$. This is the case where not many vertices qualified for scale j_{crit} can be found outside $\Pi_{i,j_{\mathsf{crit}}} \cup \Pi_{i,j_{\mathsf{crit}}-1}$. In this case, we have $\sum_{v \in \overline{\Pi}_{i,j_{\mathsf{crit}}}(\mathsf{OPT})} \mathbb{P}[C_v \leq \gamma^i / 2^{j_{\mathsf{crit}}}] \leq T^*_{i,j_{\mathsf{crit}}}$. It follows from Markov's inequality that with probability at least 0.9, OPT finds no vertex in $V_i \setminus (\Pi_{i,j_{\mathsf{crit}}} \cup \Pi_{i,j_{\mathsf{crit}}-1})$ that is qualified for scale j_{crit}, in which case any vertex $v \in V_i \setminus (\Pi_{i,j_{\mathsf{crit}}} \cup \Pi_{i,j_{\mathsf{crit}}-1})$ selected by OPT has cost $C_v > \gamma^i / 2^{j_{\mathsf{crit}}}$. Since $T_{i,j_{\mathsf{crit}}-1} \geq Y$, it follows from Theorem 3 that with probability at least 0.5, we have $\left|\left\{v \in \Pi_{i,j_{\mathsf{crit}}} \cup \Pi_{i,j_{\mathsf{crit}}-1} : C_v \leq \gamma^i / 2^{j_{\mathsf{crit}}-1}\right\}\right| \geq Y$. Furthermore, $p^*_Y \geq 0.9$ implies that with probability at least 0.9, OPT selects Y vertices within total budget γ^i. It follows from union bound that all three events above happen with probability at least 0.2, in which case we can select Y vertices from $\Pi_{i,j_{\mathsf{crit}}} \cup \Pi_{i,j_{\mathsf{crit}}-1}$ within cost budget $2\gamma^i$. This implies that $Y_{i,j_{\mathsf{crit}}} \geq Y$.
Now we deal with the other sub-case where $T^*_{i,j_{\mathsf{crit}}} > 0.1$. This represents the situation where many vertices qualified for scale j_{crit} can be found outside $\Pi_{i,j_{\mathsf{crit}}} \cup \Pi_{i,j_{\mathsf{crit}}-1}$. Roughly, Lemma 8 implies in this case that OPT finds at most $200T^*_{i,j_{\mathsf{crit}}}$ low-cost vertices in $V_i \setminus (\Pi_{i,j_{\mathsf{crit}}} \cup \Pi_{i,j_{\mathsf{crit}}-1})$. In general, these vertices might have very tiny cost. However, we show in the following Claim 14 that increasing the cost of each one of these low-cost vertices to $\gamma^i / 2^{j_{\mathsf{crit}}}$ will increase their total cost by at most $O(\gamma^i)$. This allows us to lower bound the cost of the set of vertices selected by OPT within total budget γ^i.

▷ Claim 14. We have $200T^*_{i,j_{\mathsf{crit}}} \cdot \gamma^i / 2^{j_{\mathsf{crit}}} \leq \gamma^i / 2$.

Before proving Claim 14, we complete the proof of Lemma 13. Since $T^*_{i,j_{\mathsf{crit}}} > 0.1$, from Lemma 8 we have

$$\mathbb{P}\Big[\left|\left\{v \in \overline{\Pi}_{i,j_{\mathsf{crit}}}(\mathsf{OPT}) : C_v \leq \gamma^i / 2^{j_{\mathsf{crit}}}\right\}\right| \leq 200T^*_{i,j_{\mathsf{crit}}}\Big] \;\geq\; 0.9. \tag{11}$$

Since $T_{i,j_{\mathsf{crit}}-1} \geq Y$, it follows from Theorem 19 that

$$\mathbb{P}\Big[\big|\{v \in \Pi_{i,j_{\mathsf{crit}}} \cup \Pi_{i,j_{\mathsf{crit}}-1} : C_v \leq \gamma^i/2^{j_{\mathsf{crit}}-1}\}\big| \geq Y \Big] \;\geq\; 0.5. \tag{12}$$

Together with the assumption that $p^*_{i,Y} \geq 0.9$, we have from union bound that with probability at least 0.2, both events in (11) and (12) hold and that OPT selects at least Y vertices within total budget γ^i. When all three events happen, we can replace $Y - |\Pi_{i,j_{\mathsf{crit}}}(\mathsf{OPT})|$ vertices in $\overline{\Pi}_{i,j_{\mathsf{crit}}}(\mathsf{OPT})$ by $Y - |\Pi_{i,j_{\mathsf{crit}}}(\mathsf{OPT})|$ vertices in $(\Pi_{i,j_{\mathsf{crit}}} \cup \Pi_{i,j_{\mathsf{crit}}-1}) \setminus \Pi_{i,j_{\mathsf{crit}}}(\mathsf{OPT})$ that are qualified for scale $j_{\mathsf{crit}} - 1$. Claim 14 together with (11) imply that after such replacements, we reach a subset of Y vertices in $\Pi_{i,j_{\mathsf{crit}}} \cup \Pi_{i,j_{\mathsf{crit}}-1}$ with total cost at most $2(\gamma^i + \gamma^i/2) = 3\gamma^i$. So we conclude that with probability at least 0.2, we can select at least Y vertices from $\Pi_{i,j_{\mathsf{crit}}} \cup \Pi_{i,j_{\mathsf{crit}}-1}$ within cost budget $3\gamma^i$. This implies that $Y_{i,j_{\mathsf{crit}}} \geq Y$ and finishes the proof of Lemma 13. ◀

Now we are left to prove Claim 14.

Proof of Claim 14. We consider the tour $\Pi_{i,j_{\mathsf{crit}}}$. Denote respectively $\Pi'_{i,j_{\mathsf{crit}}}(\mathsf{OPT}) \subseteq \Pi_{i,j_{\mathsf{crit}}}$ and $\overline{\Pi}'_{i,j_{\mathsf{crit}}}(\mathsf{OPT}) \subseteq V_i \setminus \Pi_{i,j_{\mathsf{crit}}}$ the (random) set of vertices visited by OPT inside and outside $\Pi_{i,j_{\mathsf{crit}}}$ within total budget γ^i. Since $T_{i,j_{\mathsf{crit}}} < Y$, it follows from Theorem 3 that

$$\mathbb{P}\Big[\big|\{v \in \Pi_{i,j_{\mathsf{crit}}} : C_v \leq \gamma^i/2^{j_{\mathsf{crit}}}\}\big| \leq Y \Big] \;\geq\; 0.5. \tag{13}$$

Since $T^*_{i,j_{\mathsf{crit}}} > 0.1$, Lemma 8 gives

$$\mathbb{P}\Big[\Big|\Big\{v \in \overline{\Pi}'_{i,j_{\mathsf{crit}}}(\mathsf{OPT}) : C_v \leq \gamma^i/2^{j_{\mathsf{crit}}}\Big\}\Big| \leq 200 T^*_{i,j_{\mathsf{crit}}} \Big] \;\geq\; 0.9. \tag{14}$$

Lemma 9 followed by Chernoff bound gives

$$\mathbb{P}\Big[\big|\{v \in \Pi_{i,j_{\mathsf{crit}}} \setminus \Pi'_{i,j_{\mathsf{crit}}}(\mathsf{OPT}) : C_v \leq \gamma^i/2^{j_{\mathsf{crit}}}\}\big| \geq 600 T^*_{i,j_{\mathsf{crit}}} \Big] \;\geq\; 0.9. \tag{15}$$

From the assumption that $p^*_{i,Y} \geq 0.9$, we have

$$\mathbb{P}\Big[\{\mathsf{OPT} \text{ selects at least } Y \text{ vertices within total budget } \gamma^i\} \Big] \;\geq\; 0.9. \tag{16}$$

By union bound, all four events in (13)-(16) happen with positive probability, in which case OPT selects at least $400 T^*_{i,j_{\mathsf{crit}}}$ vertices that are not qualified for scale j_{crit} within total budget γ^i. This implies that $400 T^*_{i,j_{\mathsf{crit}}} \cdot \gamma^i/2^{j_{\mathsf{crit}}} \leq \gamma^i$ from which Claim 14 immediately follows. ◁

5.3 The Dynamic Programming Sub-procedure $\mathsf{ALG}_{\mathsf{DP}}$

In this section, we give our dynamic program $\mathsf{ALG}_{\mathsf{DP}}$ and prove Lemma 11 restated below for convenience.

▶ **Lemma 11.** *For any phase $i \geq 0$ and any scale $j \in \{0, \cdots, \ell_i\}$, the approximate value $\widetilde{Y}_{i,j}$ computed in Step 15 of $\mathsf{ALG}_{\mathit{Stoch\text{-}Cost}}$ satisfies that (1) $\widetilde{Y}_{i,j} \geq Y_{i,j}$, and (2) $\widetilde{Y}_{i,j}$ vertices can be selected from $\Pi_{i,j} \cup \Pi_{i,j-1}$ within cost budget $6\gamma^i$ with probability at least 0.2.*

Recall, our dynamic program $\mathsf{ALG}_{\mathsf{DP}}$ is used to approximately compute the maximum number of vertices that can be selected from a certain tour with probability at least 0.2 within cost budget $3\gamma^i$. Essentially, $\mathsf{ALG}_{\mathsf{DP}}$ solves the following sub-problem: We are given a target $T \in \mathbb{Z}_{\geq 0}$, a budget $B \geq 0$ and n independent non-negative stochastic costs. The goal is to find the probability $P_{T,B}$ that there exists a subset S of size T with sum of its costs at most B. Since this general problem is NP-hard, we give a dynamic program that finds something between $P_{T,B}$ and $P_{T,2B}$. Lemma 11 follows immediately from the following Lemma 15.

▶ **Lemma 15.** *Given n independent non-negative random variables $V = \{C_1, C_2, ..., C_n\}$, a target $T \in \mathbb{Z}_{\geq 0}$ and a budget $B \geq 0$. Let $P_{T,B}$ denote the probability that there exists a subset $S \subseteq V$ of size at least T and $\sum_{i \in S} C_i \leq B$. Then there's an efficient dynamic programming* $\mathsf{ALG}_{DP}(T, B, V)$ *that outputs a value $\widetilde{P}_{T,B}$ s.t. $P_{T,B} \leq \widetilde{P}_{T,B} \leq P_{T,2B}$.*

Proof of Lemma11. We first prove property (1). Recall that $Y_{i,j}$ is the maximum number of vertices that can be selected from $\Pi_{i,j} \cup \Pi_{i,j-1}$ with probability at least 0.2 within cost budget $3\gamma^i$, and $\widetilde{Y}_{i,j}$ is the largest integer such that $\mathsf{ALG}_{\mathsf{DP}}(\widetilde{Y}_{i,j}, 3\gamma^i, \Pi_{i,j} \cup \Pi_{i,j-1}) \geq 0.2$. Consider the set of stochastic costs in $\Pi_{i,j} \cup \Pi_{i,j-1}$. By definition, we have $P_{Y_{i,j},3\gamma^i} \geq 0.2$. It follows from Lemma 15 that $\mathsf{ALG}_{\mathsf{DP}}(Y_{i,j}, 3\gamma^i, \Pi_{i,j} \cup \Pi_{i,j-1}) \geq 0.2$. This implies that $\widetilde{Y}_{i,j} \geq Y_{i,j}$ and proves property (1).

Now we prove property (2). Applying Lemma 15 we have $P_{\widetilde{Y}_{i,j},6\gamma^i} \geq \mathsf{ALG}_{\mathsf{DP}}(\widetilde{Y}_{i,j}, 3\gamma^i, \Pi_{i,j} \cup \Pi_{i,j-1}) \geq 0.2$. It follows that $\widetilde{Y}_{i,j}$ vertices can be selected from $\Pi_{i,j} \cup \Pi_{i,j-1}$ within cost budget $6\gamma^i$ with probability at least 0.2. This establishes property (2) and finishes the proof of Lemma 11. ◀

Proof of Lemma 15. We begin by discretizing each C_i to be $\overline{C}_i := \lfloor C_i \cdot n/B \rfloor \in \mathbb{N}$ and define $\overline{P}_{T,n}$ the probability that there exists a subset $S \subseteq V$ s.t. $|S| \geq T$ and $\sum_{i \in S} \overline{C}_i \leq n$. Notice that $\sum_{i \in S} \overline{C}_i \leq n$ implies that $\sum_{i \in S} C_i \leq 2B$. Therefore we have $P_{T,B} \leq \overline{P}_{T,n} \leq P_{T,2B}$. In the following, we give a dynamic programming $\mathsf{ALG}_{\mathsf{DP}}$ that computes the value $\overline{P}_{T,n}$ and we will set $\widetilde{P}_{T,B}$ in the statement of the lemma to be $\overline{P}_{T,n}$. Assume without loss of generality that $\overline{C}_i \leq n+1$ as one can truncate the distribution of $\overline{C}_i$ at $(n+1)$ without changing $\overline{P}_{T,n}$.

Denote $A(i, j)$ the jth smallest value among the first i random variables. We build a DP table where each entry $P[i, j, \ell, m]$ (for $i \in \{1, \cdots, n\}, j \in \{1, \cdots, T\}$, $\ell \in \{0, \cdots, n\}$ and $m \in \{0, \cdots, n\}$) denotes the probability that the smallest j values among the first i random variables sum up to ℓ and the jth smallest value among the first i random variables is equal to m, i.e. $\sum_{s=1}^{j} A[i, s] = \ell$ and $A[i, j] = m$.

Initial values: $\mathsf{ALG}_{\mathsf{DP}}$ initializes certain entries of the DP table as follows.

- **Case 1 (impossible events):** set $P[i, j, \ell, m]$ to be 0 if $j > i$, $m > \ell$ or $m \cdot j < \ell$.
- **Case 2 ($j = 1$):** set $P[i, 1, \ell, m] = \prod_{s \in [i]} \mathbb{P}[\overline{C}_s \geq \ell] - \prod_{s \in [i]} \mathbb{P}[\overline{C}_s > \ell]$ if $\ell = m$, and 0 otherwise.

Note that all the entries corresponding to $i = 1$ are already included in the two cases above.

Recursion: $\mathsf{ALG}_{\mathsf{DP}}$ uses the following recursion.

$$\begin{aligned} P[i, j, \ell, m] &= \mathbb{P}[\overline{C_i} > m] \cdot P[i-1, j, \ell, m] + \sum_{u=0}^{m-1} \mathbb{P}[\overline{C_i} = u] \cdot P[i-1, j-1, \ell-u, m] \\ &+ \mathbb{P}[\overline{C_i} = m] \cdot \Big(\sum_{u=0}^{m} P[i-1, j-1, \ell-m, m-u] - \sum_{u=1}^{m} P[i-1, j, \ell-u, m-u] \Big). \end{aligned} \tag{17}$$

Output: after computing all the entries of the DP table, $\mathsf{ALG}_{\mathsf{DP}}$ outputs $\overline{P}_{T,n}$ that equals $\sum_{\ell=0}^{n} \sum_{m=0}^{\ell} P[n, T, \ell, m]$.

Now we prove the correctness of $\mathsf{ALG}_{\mathsf{DP}}$. Given the definition of $P[i, j, \ell, m]$, we can immediately verify that the assignment of initial values and the final output are correct if all the entries of the DP table computed from (17) are also correct. To see the correctness of the recursion, we consider the outcome of $\overline{C}_i$. When $\overline{C}_i > m$, in order to satisfy $\sum_{s=1}^{j} A[i, s] = \ell$

and $A[i,j] = m$, one must have that $\overline{C}_i$ is not in the j smallest values among the first i random variables. This verifies the first term in (17). When $\overline{C}_i = u < m$, in order to satisfy $\sum_{s=1}^{j} A[i,s] = \ell$ and $A[i,j] = m$, one must have that $\overline{C}_i$ is one of the jth smallest values among the first i random variables. Also notice that in this case, the $(j-1)$th smallest value among the first $(i-1)$ random variables is still m and that $\sum_{s=1}^{j-1} A[i-1,s] = \ell - \overline{C}_i$. This gives the second term in (17).

Now we verify the last term in (17, which corresponds to the case where $\overline{C}_i = m$. In this case, we might as well select $\overline{C}_i$ as one of the j smallest values among the first i random variables. In order to satisfy $\sum_{s=1}^{j} A[i,s] = \ell$ and $A[i,j] = m$, we need the smallest $(j-1)$ values among the first $(i-1)$ random variables to sum up to $\ell - m$ and the $(j-1)$th smallest value to be at most m, i.e. $\sum_{s=1}^{j-1} A[i,s] = \ell - m$ and $A[i,j-1] \leq m$. The probability of this event is exactly $\sum_{u=0}^{m} P[i-1, j-1, \ell-m, m-u]$. However, in order to ensure that $A[i,j] = m$, we also need the jth smallest value among the first $(i-1)$ random variables to be at least m (i.e. $A[i-1,j] \geq m$) and the outcomes that don't satisfy this condition needs to be excluded from the previous event. Putting everything together, we have the following:

$$\begin{aligned}
&\mathbb{P}\Big\{\sum_{s=1}^{j-1} A[i-1,s] = \ell - m, A[i-1,j-1] \leq m, A[i-1,j] \geq m\Big\} \\
&= \mathbb{P}\Big\{\sum_{s=1}^{j-1} A[i-1,s] = \ell - m, A[i-1,j-1] \leq m\Big\} \\
&\qquad - \mathbb{P}\Big\{\sum_{s=1}^{j-1} A[i-1,s] = \ell - m, A[i-1,j-1] \leq m, A[i-1,j] < m\Big\} \\
&= \sum_{u=0}^{m} P[i-1, j-1, \ell-m, m-u] - \sum_{u=1}^{m} P[i-1, j, \ell-u, m-u].
\end{aligned}$$

This immediately gives the last term in (17) and finishes the proof of Lemma 15. ◀

References

1 Sanjeev Arora and George Karakostas. 2+ ε approximation algorithm for the k-MST problem. In *11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 754–759. SIAM, 2000.

2 Sunil Arya and Hariharan Ramesh. A 2.5-factor approximation algorithm for the k-MST problem. *Information Processing Letters*, 65(3):117–118, 1998.

3 Arash Asadpour and Hamid Nazerzadeh. Maximizing stochastic monotone submodular functions. *Management Science*, 62(8):2374–2391, 2016.

4 Baruch Awerbuch, Yossi Azar, Avrim Blum, and Santosh Vempala. New Approximation Guarantees for Minimum-Weight k-Trees and Prize-Collecting Salesmen. *SIAM J. Comput.*, 28(1):254–262, 1998.

5 Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. When LP Is the Cure for Your Matching Woes: Improved Bounds for Stochastic Matchings. *Algorithmica*, 63(4):733–762, 2012.

6 Nikhil Bansal and Viswanath Nagarajan. On the Adaptivity Gap of Stochastic Orienteering. In *IPCO*, pages 114–125, 2014.

7 Anand Bhalgat, Ashish Goel, and Sanjeev Khanna. Improved Approximation Results for Stochastic Knapsack Problems. In *SODA*, pages 1647–1665, 2011.

8 Avrim Blum, R. Ravi, and Santosh Vempala. A Constant-factor Approximation Algorithm for the k MST Problem (Extended Abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 442–448, 1996.

9 Domagoj Bradac, Sahil Singla, and Goran Zuzic. (Near) Optimal Adaptivity Gaps for Stochastic Multi-Value Probing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 49:1–49:21, 2019.

10 Chandra Chekuri, Nitish Korula, and Martin Pál. Improved algorithms for orienteering and related problems. *ACM Transactions on Algorithms (TALG)*, 8(3):23, 2012.

11 Brian C. Dean, Michel X. Goemans, and Jan Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 208–217. IEEE, 2004.

12 Amol Deshpande, Lisa Hellerstein, and Devorah Kletenik. Approximation Algorithms for Stochastic Submodular Set Cover with Applications to Boolean Function Evaluation and Min-Knapsack. *ACM Trans. Algorithms*, 12(3):42:1–42:28, 2016.

13 Alina Ene, Viswanath Nagarajan, and Rishi Saket. Approximation Algorithms for Stochastic k-TSP. In *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

14 David A Freedman. On tail probabilities for martingales. *the Annals of Probability*, 3(1):100–118, 1975.

15 Hao Fu, Jian Li, and Pan Xu. A PTAS for a Class of Stochastic Dynamic Programs. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

16 N. Garg. A 3-approximation for the Minimum Tree Spanning K Vertices. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, FOCS '96, pages 302–, 1996.

17 Naveen Garg. Saving an epsilon: a 2-approximation for the k-MST problem in graphs. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 396–402. ACM, 2005.

18 Michel X. Goemans and Jan Vondrák. Stochastic Covering and Adaptivity. In *LATIN 2006: Theoretical Informatics, 7th Latin American Symposium, Proceedings*, pages 532–543, 2006.

19 Sudipto Guha and Kamesh Munagala. Multi-armed Bandits with Metric Switching Costs. In *ICALP*, pages 496–507, 2009.

20 Sudipto Guha and Kamesh Munagala. Adaptive uncertainty resolution in bayesian combinatorial optimization problems. *ACM Transactions on Algorithms (TALG)*, 8(1):1, 2012.

21 Anupam Gupta, Haotian Jiang, Ziv Scully, and Sahil Singla. The Markovian Price of Information. In *Integer Programming and Combinatorial Optimization, IPCO*, pages 233–246, 2019.

22 Anupam Gupta, Ravishankar Krishnaswamy, Marco Molinaro, and R. Ravi. Approximation Algorithms for Correlated Knapsacks and Non-martingale Bandits. In *FOCS*, pages 827–836, 2011.

23 Anupam Gupta, Ravishankar Krishnaswamy, Viswanath Nagarajan, and R. Ravi. Approximation Algorithms for Stochastic Orienteering. In *SODA*, 2012. URL: `http://dl.acm.org/citation.cfm?id=2095116.2095237`.

24 Anupam Gupta and Viswanath Nagarajan. A Stochastic Probing Problem with Applications. In *IPCO*, pages 205–216, 2013.

25 Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Adaptivity Gaps for Stochastic Probing: Submodular and XOS Functions. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1688–1702. SIAM, 2017.

26 Zhihao Jiang and Haoyu Zhao. An FPTAS for Stochastic Unbounded Min-Knapsack Problem. In *International Workshop on Frontiers in Algorithmics*, pages 121–132. Springer, 2019.

27 Kumar Jogdeo and Stephen M Samuels. Monotone convergence of binomial probabilities and a generalization of Ramanujan's equation. *The Annals of Mathematical Statistics*, 39(4):1191–1195, 1968.

28 Robert Kleinberg, Bo Waggoner, and Glen Weyl. Descending Price Optimally Coordinates Search. *arXiv preprint*, 2016. `arXiv:1603.07682`.

29 Will Ma. Improvements and Generalizations of Stochastic Knapsack and Multi-Armed Bandit Approximation Algorithms: Extended Abstract. In *SODA*, pages 1154–1163, 2014.

30 Sridhar Rajagopalan and V Vazirani. Logarithmic approximation of minimum weight k trees. *Unpublished manuscript*, 1995.

31 Ramamurthy Ravi, Ravi Sundaram, Madhav V Marathe, Daniel J Rosenkrantz, and Sekharipuram S Ravi. Spanning trees-short or small. *SIAM Journal on Discrete Mathematics*, 9(2):178–200, 1996.

32 Sahil Singla. *Combinatorial Optimization Under Uncertainty: Probing and Stopping-Time Algorithms.* PhD thesis, Carnegie Mellon University, 2018.

33 Sahil Singla. The Price of Information in Combinatorial Optimization. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms.* SIAM, 2018.

34 Martin L. Weitzman. Optimal search for the best alternative. *Econometrica: Journal of the Econometric Society*, pages 641–654, 1979.

A Missing Proofs in Section 2

▶ **Lemma 6** (Bi-criteria Orienteering). *There is an efficient algorithm* $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ *that finds a tour of length* $O(1)\cdot B$ *while collecting at least* $(\mathsf{OPT}_{\mathsf{Orient}}-\epsilon)$ *profit, where* $\epsilon = 1/\mathrm{poly}(n)$ *can be made arbitrarily small.*

Proof of Lemma 6. Assume without loss of generality that $\mathsf{OPT}_{\mathsf{Orient}} > 0$. Denote $\rho = 3$ the approximation factor of k-TSP algorithm $\mathsf{ALG}_{k\text{-}\mathsf{TSP}}$ from [8]. Denote $R_{\max} := \max_{v\in V} R_v$ and $R_{\min} := \min_{v\in V} R_v$ the maximum and minimum profit in the Orienteering instance. Notice, $\mathsf{OPT}_{\mathsf{Orient}} \in [R_{\min}, n\cdot R_{\max}]$.

$\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ applies binary search in $[R_{\min}, n\cdot R_{\max}]$, starting with profit target $(R_{\min} + n\cdot R_{\max})/2$. For each profit target λ, $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ runs $\mathsf{ALG}_{k\text{-}\mathsf{TSP}}$ with target reward λ to obtain a tour Π_λ whose length is denoted as $\ell(\Pi_\lambda)$. $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ performs binary search over $\lambda \in [R_{\min}, n\cdot R_{\max}]$ until finding two values $\lambda_l < \lambda_h \le \lambda_l + \epsilon$ such that $\ell(\Pi_{\lambda_l}) \le \rho B$ and $\ell(\Pi_{\lambda_h}) > \rho B$, in which case $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ returns the tour Π_{λ_l}. Here we assumed without loss of generality that $\ell(\Pi_{n\cdot R_{\max}}) > \rho B$ as otherwise $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ can simply return the tour $\Pi_{n\cdot R_{\max}}$. Notice that $\ell(\Pi_{\lambda_h}) > \rho B$ implies that $\mathsf{OPT}_{\mathsf{Orient}} < \lambda_h$ and therefore $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ finds reward at least $\lambda_l \ge \lambda_h - \epsilon > \mathsf{OPT}_{\mathsf{Orient}} - \epsilon$. The length of the tour found by $\mathsf{ALG}_{\mathsf{Bicrit\text{-}Orient}}$ is $\ell(\Pi_{\lambda_l}) \le \rho B = O(1)\cdot B$. This finishes the proof of Lemma 6. ◀

B Missing Proofs in Section 3

▶ **Lemma 7** (Key Lemma). *If for some universal constants* $C > 0$, $\gamma > 1$, *any phase* $i \ge 1$, *and any possible* σ_{i-1}, *the algorithm* $\mathsf{ALG}_{\mathsf{Meta}}$ *satisfies*

$$u_i(\sigma_{i-1}) \le C\cdot u_i^*(\sigma_{i-1}) + \frac{u_{i-1}(\sigma_{i-1})}{\gamma^2},$$

then $\mathsf{ALG}_{\mathsf{Meta}}$ *is a non-adaptive* $O(1)$*-approximation algorithm.*

Proof of Lemma 7. For any phase $i \ge 0$, denote u_i the probability that $\mathsf{ALG}_{\mathsf{Meta}}$ enters phase $i+1$ in the probing stage and u_i^* the probability that OPT has cost more than γ^i. Taking expectation over σ_{i-1} for the pre-condition of Lemma 7, we have $u_i \le C\cdot u_i^* + u_{i-1}/\gamma^2$. It follows that

$$\sum_{i\ge1} u_i\cdot\gamma^i \;\le\; C\cdot\sum_{i\ge1} u_i^*\cdot\gamma^i + u_0/\gamma + 1/\gamma\cdot\sum_{i\ge1} u_i\cdot\gamma^i,$$

which gives

$$(1-1/\gamma)\cdot\sum_{i\geq 1} u_i\cdot\gamma^i \;\leq\; O(1)\cdot\sum_{i\geq 1} u_i^*\cdot\gamma^i + 1/\gamma. \tag{18}$$

We also notice that

$$\mathsf{OPT} \quad\geq\quad \sum_{i\geq 0}(u_i^* - u_{i+1}^*)\cdot\gamma^i \quad=\quad (1-1/\gamma)\cdot\sum_{i\geq 1} u_i^*\cdot\gamma^i + 1,$$

and that

$$\mathsf{ALG}_{\mathsf{Meta}} \quad\leq\quad O(1)\cdot\sum_{i\geq 0}(u_i - u_{i+1})\cdot\gamma^{i+1} \quad=\quad O(1)\cdot\sum_{i\geq 0} u_i\cdot\gamma^i.$$

It follows from (18) that $\mathsf{ALG}_{\mathsf{Meta}} \leq O(1)\cdot\mathsf{OPT}$. This finishes the proof of Lemma 7. ◀

C Missing Proofs in Section 5

▶ **Lemma 12.** *For $\gamma = 1.1$, any phase $i > 0$ in the probing stage of Stoch-Cost k-TSP satisfies*

$$u_i(\sigma_{i-1}) \leq 100u_i^*(\sigma_{i-1}) + \frac{u_{i-1}(\sigma_{i-1})}{\gamma^2}. \tag{7}$$

Proof of Lemma 12. We fix any outcome σ_{i-1} of vertices visited by $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$ in the first $i-1$ phases of its probing stage. The lemma trivially holds in the case where $u_i^*(\sigma_{i-1}) \geq 0.01$ as we have $100u_i^*(\sigma_{i-1}) \geq 1$. If $u_{i-1}(\sigma_{i-1}) = 0$ which means that $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$ already selects k vertices before entering phase i in the probing stage, then $u_i(\sigma_{i-1}) = 0$ and again the lemma trivially holds. We therefore assume that $u_i^*(\sigma_{i-1}) < 0.01$ and that $u_{i-1}(\sigma_{i-1}) = 1$. Now proving Lemma 12 is equivalent to proving

$$u_i(\sigma_{i-1}) \leq 100u_i^*(\sigma_{i-1}) + 1/\gamma^2. \tag{19}$$

Denote $k(\sigma_{i-1})$ the remaining target at the beginning of phase i in the probing stage of $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$. We first consider Selection-Process 1 and denote $N_{\mathsf{old}}(\sigma_{i-1})$ the number of vertices selected from σ_{i-1} in this process. We assume without loss of generality that $N_{\mathsf{old}}(\sigma_{i-1}) < k(\sigma_{i-1})$, as otherwise our algorithm has already selected k vertices after Selection-Process 1 and (19) immediately follows. Since Selection-Process 1 uses cost budget γ^i to select as many unselected vertices from σ_{i-1} as possible, OPT can select at most $N_{\mathsf{old}}(\sigma_{i-1}) + k - k(\sigma_{i-1})$ vertices from σ_{i-1} within total budget γ^i. Denote $N_{\mathsf{new}}(\sigma_{i-1}) := k(\sigma_{i-1}) - N_{\mathsf{old}}(\sigma_{i-1})$ the remaining target for $\mathsf{ALG}_{\mathsf{Stoch\text{-}Cost}}$ after Selection-Process 1. It follows that in order to select k vertices within total budget γ^i, OPT needs to select at least $N_{\mathsf{new}}(\sigma_{i-1})$ vertices from V_i within total budget γ^i. Therefore, $u_i^*(\sigma_{i-1}) < 0.01$ implies that OPT selects at least $N_{\mathsf{new}}(\sigma_{i-1})$ vertices from V_i within total budget γ^i with probability at least 0.99. Applying Lemma 13 with $Y = N_{\mathsf{new}}(\sigma_{i-1})$, it follows that our algorithm finds at least $N_{\mathsf{new}}(\sigma_{i-1})$ vertices from V_i which can be selected within cost budget $6\gamma^i$ with probability at least 0.2. This implies that $u_i(\sigma_{i-1}) \leq 0.8 \leq 1/\gamma^2$ and (19) is established. ◀