# Various Approaches to the Stochastic K-Server and Stacker-Crane Problems

Alexander D. Friedman

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Masters of Science
in
Mathematics

Joseph A. Ball, Chair
Sharath Raghvendra
John Rossi

May 04, 2017
Blacksburg, Virginia

Keywords: K-Server Problem, Stacker-Crane Problem, K-Median, Zoning Algorithm, Probabilistic Hyperoctree, Voronoi Simplicial Decomposition

# Various Approaches to the Stochastic K-Server and Stacker-Crane Problems

Alexander D. Friedman

(ABSTRACT)

The $k$-Server and Stacker-Crane problems are of significant importance to both the theoretical and applied computer science communities. In this thesis we present algorithms for the stochastic $k$-server problem under both known and unknown distribution models, providing bounds on expected cost. We give a lookahead algorithm for special geometric variants of the problem. Finally, we analyze two fast approximation algorithms to the stochastic Stacker-Crane problem including both theoretical and experimental results.

The work was partially supported through the National Science Foundation (NSF) grant 1464276.

# Various Approaches to the Stochastic K-Server and Stacker-Crane Problems

Alexander D. Friedman

(GENERAL AUDIENCE ABSTRACT)

In recent years there has been a trend towards large-scale logistics for individual members of the public, such as ride-sharing services and drone package delivery. Efficient coordination of pickups and deliveries is essential in order to keep costs and wait times down.

In this thesis we present these types of problems in a more general framework, expanding applicability of our discussion to an even wider domain of problems. We present fast new algorithms with supporting theoretical and experimental analysis, providing certain guarantees about how close our algorithms compare to a theoretically optimal approach.

The work was partially supported through the National Science Foundation (NSF) grant 1464276.

# Dedication

I dedicate this thesis to Neville: a grandfather of seemingly infinite patience who helped foster a passion for mathematics in my youth.

# Acknowledgments

In addition to my entire committee, this work would not have been possible without a joint collaboration with Dr. Abhijin Adiga and Dr. Sharath Raghvendra on stochastic $k$-Server algorithms.

Further acknowledgments of the National Science Foundation and their ever essential mission in promoting progress of science and engineering, and advancing research and education.

Finally, acknowledgments of the many professors at the Virginia Tech Mathematics Department that leave their doors open for discussion, debate, and guidance in the most fascinating of academic fields.

# Contents

# List of Algorithms



# List of Figures

# Chapter 1

# Introduction

Throughout the last century there has been a great deal of interest in solving and analyzing the behavior of numerous different assignment problems with varying levels of abstraction. Famous assignment problems, such as the Traveling Salesman Problem and $k$-Server Problem, have been studied extensively in deterministic and randomized settings due to their importance to a wide array of different applications, from geometric task planning to memory allocation within a computer.

In more recent years large-scale logistics have become utterly essential to the way our very society functions. From enormous delivery networks and food distribution to ride-sharing services and even internet traffic, we are becoming increasingly reliant on fast and efficient services. With the promise of drone delivery and self-driving (perhaps even owner less) vehicle future on the horizon; with booming populace and massive increases of city-dwelling all around the world; it is difficult to see an end to the insatiable desire for faster, cheaper and better methods. In light of this necessity, it is essential to provide algorithms that both perform well and serve as a basis of, or at least give guidance towards, the metrical algorithms of the future.

In this thesis we concentrate on two problems of interest: the $k$-Server Problem and the Stacker-Crane Problem. In each case we consider algorithms running in a stochastic setting, i.e., where points are are generated i.i.d according to some distribution.

We present algorithms for the $k$-Server problem for the known and unknown distribution models, in a general pseudo-metric setting (chapter 3). We then provide a work-function-inspired heuristic algorithm that may be applied when the domain is a subset of $\mathbb{R}^d$, with primary concentration on computation in a uniformly distributed setting with $L_1$ and $L_2$ metrics (chapter 4).

We also present two algorithms, ASPLICE and PLG, that provide approximations to the Stochastic Stacker-Crane Problem on compact subsets of $\mathbb{R}^d$. Under certain conditions we show that the former is asymptotically almost-surely near-optimal, and under further con-

ditions the latter may also be shown to be, whilst running in linear time (chapter 5).

In chapter 2 we introduce the $k$-Server and Stacker-Crane Problem, providing background motivation and a brief summary of a few well-known results.

# Chapter 2

# Considered Problems

## 2.1 $k$-Server Problem

There are many ways to approach the topic of the $k$-server problem, which is in large part due to both the breadth of related research and its numerous applications. We will begin by discussing a special case of the $k$-server problem, the **paging problem**, before giving the more general problem.

In the development of computers there has generally been a push for both faster and larger memory machines. One issue that began to emerge in the 60s was that the reading and writing to memory took far more time than any other operation and, by the 80s, this became a design bottleneck. The solution was to use cache "fast" memory, that was more expensive but faster than disk "slow" memory.

The "fast" memory was so much faster that reading and writing to it were considered negligible operations. The idea of paging was to check the "fast" memory for a copy of some data and if it were not present then the data would be copied in from "slow" memory. As there is limited space, a decision must be made as to which block of "fast" memory should be replaced.

This gives us the paging problem: *Given a sequence of memory requests, how do we choose the order of memory overwrites so as to minimize the number of "slow" reads?*

An ambiguity emerges that needs to be resolved, specifically, do we know all memory requests ahead of time or are we receiving them one at a time? We define two different types of algorithms: **offline** algorithms and **online** algorithms, where the former knows all requests ahead of time.

Suppose we have 2 memory blocks of "fast" memory and 3 of "slow". A deterministic online algorithm may always be forced to pay the full time cost at every request. This may be done

by considering whatever is loaded in the "fast" memory and requesting the memory that isn't loaded. Without the ability to look ahead against such sequence of requests, we lose any benefit from having "fast" memory.

This deterministic model could be considered overly restrictive. In many ways it is unrealistic to assume that the requests are being made with the intention of costing the algorithm a lot, and so there are numerous other types of analysis, allowing for different modes of randomness and robustness, as discussed in appendix B.1.

Though a well-studied problem on its own, the paging problem lacks the kind of generality that makes the $k$-server problem universal, indeed we will show that the paging problem is just a special case.

In order to begin we will need some notation. Recall that a pseudo-metric space is a space of points $X$ with $d : X \times X \to [0, \infty)$ that satisfies $d(x, y) \leq d(x, z) + d(z, y)$, $d(x, x) = 0$, and $d(x, y) = d(y, x)$. This differs from a metric space in that it does not require $d(x, y) = 0$ to imply $x = y$.

**Definition 2.1.** *Fix some integer $k > 0$ and let $(X, d)$ be a pseudo-metric space. We define a **centered partition** to be a tuple $(\mathcal{C}, \mathbf{Z})$, where $\mathbf{Z} = \{X_1, \ldots, X_k\}$ is a partition of $X$ into **zones** and where $\mathcal{C} = \{c_1, \ldots, c_k\} \subseteq X$ is a **configuration of centers** with the property that one center is in each zone, i.e., $c_i \in X_i$ for $i = 1, \ldots, k$.*

*We define a distance from a request $x$ with respect to a configuration to be the distance to the corresponding center of the zone that $x$ falls into,*

$$d(x, \mathcal{C}) = d(x, c_i), \quad \text{where } x \in X_i \in \mathbf{Z}$$

*We also define a distance between configurations $\mathcal{C}_1, \mathcal{C}_2$, which is given by the minimum-weight perfect matching. In other words, it is the minimum-cost bijection from $\mathcal{C}_1$ to $\mathcal{C}_2$ where the cost of the bijection is taken between mapped centers, i.e., $c_i \mapsto \hat{c}_k$ then $d(c_i, \hat{c}_k)$ is incurred.*

**Definition 2.2** ($k$-Server Problem)**.** *Given some sequence of requests $S = (x_1, \ldots, x_n)$ and an initial centered partition $(\mathcal{C}_0, \mathbf{Z}_0)$, a candidate solution is a sequence of centered partitions $((\mathcal{C}_t, \mathbf{Z}_t))_{t=1}^n$ where $x_t \in \mathcal{C}_t$ for $t = 1, \ldots, n$. The $k$-server problem is to find a candidate solution that minimizes*

$$\sum_{t=0}^{n-1} d(\mathcal{C}_t, \mathcal{C}_{t+1})$$

**Remark 2.3.** *Suppose we let $k$ be the number of memory blocks in cache and let $X$ be a space of memory, and use the discrete metric $d(x, y)$ which is $1$ if $x \neq y$, otherwise it is $0$. Hence when requesting data, if the request is already in cache (and therefore one of the centers), then we pay no cost as it is "fast" memory, whereas if it is not one of the centers then it must become one of the centers, incurring the cost of $1$. Therefore the paging problem just becomes about finding a sequence of centers that minimizes the total cost of servicing memory requests.*

There are two well known conjectures about the $k$-server problem, as stated in [26]:

**Conjecture 2.4** (The $k$-server conjecture)**.** *For every metric space with more than $k$ distinct points, the competitive ratio of the $k$-server problem is exactly $k$ (see appendix B.1).*

**Conjecture 2.5** (The randomized $k$-server conjecture)**.** *For every metric space, there is a randomized online algorithm for the $k$-server problem with competitive ratio $O(\log k)$.*

It has been shown that $k$ is a lower bound for any metric space with at least $k+1$ points [30], hence confirming a stronger form of the first conjecture. Also for the second conjecture $\Omega(\log k)$ is known to be an lower bound and some $O(\log k)$ are known for the paging problem [15, 31].

There are numerous algorithms that have been introduced for specific metrics, but some of the more commonly known are

1. **Greedy Algorithm** Deterministic online algorithm with unbounded competitive ratio. Moves the closest center to each request [26].
2. **Retrospective Algorithm** Deterministic online algorithm with unbounded competitive ratio. Moves so as to minimize the $k$-server problem over the past $s$ requests [26].
3. **Work Function Algorithm** Deterministic online algorithm with a competitive ratio of at most $2k-1$ for any metric space. Minimizes the sum of a maintained work function and the reconfiguration cost $d(\mathcal{C}_{t-1}, \mathcal{C}_t)$ [26].
4. **Harmonic Algorithm** Randomized online algorithm with a proven competitive ratio of $O(2^k \log k)$ for all metric spaces, though $k(k+1)/2$ has been shown for others. There is a standing conjecture that $k(k+1)/2$ holds for all metric spaces. Moves a server to a request with a probability inversely proportional to the distance [8].
5. **Polylogarithmic Algorithmic** Randomized algorithm with competitive ratio $O(\log^2 k \log^3 n \log\log n)$ on any metric [7].

## 2.2 Stacker-Crane Problem

The Stacker-Crane Problem (SCP) has been studied almost as long as the $k$-server problem [18], serving as a generalization of the Travelling Salesman problem. Even when dealing with specific application to the Euclidean metric, the problem is $\mathcal{NP}$-hard as it is a generalization of Euclidean TSP (ETSP), which is known to be $\mathcal{NP}$-complete [34].

A simple example of SCP: *A taxi driver starts at a depot, proceeds to pickup and drop off clients, and then returns to the depot. Suppose the driver knows all pick up and destination*

*pairs ahead of time, what is the minimum distance that must be travelled to serve all requests?* Any solution to this would have clear application to robotic task scheduling and numerous other practical problems.

Numerous heuristic algorithms have been proposed, e.g., [9], [41], all with varying levels of analysis. One paper is of particular interest to us, specifically [39], in which the authors present an asymptotically almost-surely optimal solution to a special, but important, case of SCP. We will briefly summarize their algorithm after giving the formal statement of SCP.

**Definition 2.6** (Stacker-Crane Tour)**.** *For some compact $\Omega \subset \mathbb{R}^d$, given sets $\mathcal{X}_n = \{X_1, \ldots, X_n\} \subset \Omega$ and $\mathcal{Y}_n = \{Y_1, \ldots, Y_n\} \subset \Omega$, a* ***Stacker-Crane Tour*** *is a Hamiltonian cycle that contains all edges $(X_i, Y_i)$.*

**Definition 2.7** (Euclidean Stochastic Stacker-Crane Problem)**.** *Let $\Omega \subset \mathbb{R}^d$ be compact and let $\mathcal{X}_n = \{X_1, \ldots, X_n\} \subset \Omega$ and $\mathcal{Y}_n = \{Y_1, \ldots, Y_n\} \subset \Omega$ be sets of random variables distributed according to p.d.f.s, $\phi_P, \phi_D \colon \Omega \to [0, \infty)$ respectively. The* ***Euclidean Stochastic Stacker-Crane Problem (ESCP)*** *is to find a Stacker-Crane Tour with minimum cost, where the cost is given by the total Euclidean length of the tour.*

We now introduce the concept of the Euclidean Bipartite Matching Problem,

**Definition 2.8** (Euclidean Bipartite Matching Problem)**.** *Given $\mathcal{X}_n = \{x_i\}_{i=1}^n \subset \Omega_X$ and $\mathcal{Y}_n = \{y_i\}_{i=1}^n \subset \Omega_Y$, with $\Omega_X, \Omega_Y \subset \mathbb{R}^d$. The* ***Euclidean Bipartite Matching Problem****,* EBMP$(\mathcal{X}_n, \mathcal{Y}_n)$*, is to find the matching (a.k.a., permutation, finite bijection), $\sigma \in \prod_n$, that minimizes*

$$\sum_{i=1}^{n} \|x_i - y_{\sigma(i)}\|_2$$

In [39] an important relationship between ESCP and EBMP is given. In essence, adding the edges produced by the EBMP between $\mathcal{Y}_n$ and $\mathcal{X}_n$ to the set of edges $\{(X_i, Y_i)\}$, is "close" to an optimal solution to EBMP.

**input :** a set of demands $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $n > 1$.
**output:** a Stacker Crane tour through $\mathcal{S}$.

1 **initialize** $\sigma \leftarrow$ solution to Euclidean bipartite matching problem between sets $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\mathcal{Y} = \{y_1, \ldots, y_n\}$ computed by using a bipartite matching algorithm $\mathcal{M}$.
2 Add the $n$ pickups-to-delivery links $x_i \to y_i$, $i = 1, \ldots, n$.
3 Add the $n$ matching links $y_i \to x_{\sigma(i)}$, $i = 1, \ldots, n$.
4 $\{S_j\}_{i=1}^{N_n} \leftarrow$ subtours.
5 `MergeSubtours(`$\{S_j\}_{i=1}^{N_n}$`)`

**Algorithm 1:** Pseudocode for the SPLICE algorithm

Here, MergeSubtours$(\cdot)$ may be any algorithm. Trivial algorithms can perform the merge task without constraint in $\Theta(N_n)$ (where $N_n$ is the number of cycles) if the cycles are known before hand, or $\Theta(n)$ if we have to merge tours from an edge set only.

**Theorem 2.9** ( [39, Theorem 4.5]). *Let $d \geq 2$. Let $\mathcal{X}_n$ be a set of points $\{X_1, \ldots, X_n\}$ that are i.i.d. in a compact set $\Omega \subset \mathbb{R}^d$ and distributed according to a density $\phi_P$; let $\mathcal{Y}_n$ be a set of points $\{Y_1, \ldots, Y_n\}$ that are i.i.d. in a compact set $\Omega \subset \mathbb{R}^d$ and distributed according to a density $\phi_D$. Then*

$$\lim_{n \to +\infty} \frac{\operatorname{cost}\left[\operatorname{SPLICE}(\mathcal{X}_n, \mathcal{Y}_n)\right]}{\operatorname{cost}\left[\operatorname{opt}(\mathcal{X}_n, \mathcal{Y}_n)\right]} = 1, \quad a.s.$$

Finally, in [39] they give a lower bound on the cost of an optimal Stacker-Crane tour:

**Theorem 2.10** ( [39, Theorem 5.4]). *Let* $\operatorname{cost}(\operatorname{SCP}_n)$ *be the length of the optimal Stacker-Crane tour through pickup-delivery pairs $(X_1, Y_1), \ldots, (X_n, Y_n)$ distributed according to PDFs $\phi_p$ and $\phi_d$, on compact $\Omega \subset \mathbb{R}^d$, where $d \geq 2$,*

$$\liminf_{n \to +\infty} \frac{\operatorname{cost}(\operatorname{SCP}_n)}{n} \geq \mathbb{E}[d(x, y)] + d_{\mathrm{W}}(\phi_d, \phi_p)$$

**Definition 2.11** ($k$-Stacker-Crane Problem). *For some compact $\Omega \subset \mathbb{R}^d$, given sets $\mathcal{X}_n = \{X_1, \ldots, X_n\} \subset \Omega$ and $\mathcal{Y}_n = \{Y_1, \ldots, Y_n\} \subset \Omega$, the* ***$k$-Stacker-Crane Problem*** *is to find $k$-disjoint tours that minimize total combined length such that every edge $(X_i, Y_i)$ is contained in one of the tours.*

**Remark 2.12.** *We shall be using asymptotic analysis to study the Stacker-Crane Problem. We may always merge $k$ tours into a single one by adding at most $k$ edges which add a total cost of less than $k\Delta$, where $\Delta$ is the diameter of the vertex set. In [39] they show that the Stacker-Crane tour grows linearly in number of pairs $n$, hence the additive cost of merging tours is asymptotically negligible. It follows that there is no difference between the $k$-Stacker-Crane Problem and Stacker-Crane Problem in asymptotic analysis.*

# Chapter 3

# Zoning Algorithm

## 3.1 Background

**Definition 3.1.** *For a metric space $(X, d)$ with probability measure $P$, a set of $k$ points $K^* \subseteq X$, given by*

$$K^* = \underset{K \subset X, |K|=k}{\arg\min} \mathop{\mathbb{E}}_{x \in X}[d(x, K)] = \underset{K \subset X, |K|=k}{\arg\min} \int_{x \in X} d(x, K) P(\,\mathrm{d}x)$$

*is referred to as the* ***$k$-median centers of*** *$X$. Here $d(x, K)$ is taken to be the distance from $x$ to the closest point in $K$. In other words, the $k$-median centers $K^*$ is a subset that minimize the expected distance from any point to the closest center in $K$.*

**Remark 3.2.** *The $k$-median defined here is a more general probabilistic definition. The usual treatment of finite spaces is to use a discrete measure, let all points be equiprobable and make the space a multiset so that higher probabilities may be represented by repeat elements in the space. Using definition 3.1 we obtain the usual definition:*

$$K^* = \underset{K \subset X}{\arg\min} \int_{x \in X} d(x, K) P(\,\mathrm{d}x) = \frac{1}{|X|} \underset{K \subset X}{\arg\min} \sum_{x \in X} d(x, K)$$

High probability approximations of the $k$-median based on sampling has been studied, e.g., [13], though the approximation error is often dependent on $\Delta$, the diameter of the metric space. Some results and algorithms do exist that are independent of $\Delta$, e.g., [33].

**Theorem 3.3.** *For any online algorithm $\mathbb{A}$ serving requests $S$ sampled i.i.d from $(X, d, P)$, the expected cost is bounded below*

$$|S| \mathop{\mathbb{E}}_{x \in X}[d(x, K^*)] \leq \mathbb{E}[\mathrm{cost}(\mathbb{A})]$$

*where $K^*$ is the $k$-median centers.*

*Proof.* When a request arrives the algorithm must assign a server to it, so the cost of the algorithm must be at least the distance from the closest server to the request. The expected cost is therefore bounded below by the expected minimum distance from a request to the $k$-servers, $K$, i.e. $\mathbb{E}_{x\in X}[d(x, K)]$. This is in turn minimized if the servers have the configuration of the $k$-median centers $K^*$. Hence the result follow by linearity of expectation. □

## 3.2 Known Distribution

We begin with a simple online analysis assumption; specifically that the online algorithm knows both the space and the distribution from which requests are generated i.i.d. This model may seem overly specific at first, however, for many applications, e.g., delivery planning, historical data may be sufficient for this model to be applicable.

We begin by introducing the zoning algorithm (algorithm 2) that utilizes the triangle inequality in a simple manner. The core idea behind the algorithm is that the metric space $(X, d)$ can be partitioned into zones for which each has a single server. Upon receiving a request in a given zone, the corresponding server attends to it. These zones are implicitly maintained in the algorithm. The use of the $k$-median centers $K^*$, closeness to centers being the defining characteristic, allows us to relate the expected cost to theorem 3.3.

**Data:** Metric space $(X, d)$, probability distribution $P$, a sequence of requests $S = (x_1, \ldots, x_r)$, initial server positions $(c_1, \ldots, c_k)$
**Result:** Sequence of servers assigned in an online manner

```
Calculate the k-median centers K* on the metric space
Move and associate the servers to the median centers according to a minimum
  bipartite matching over their pairwise distances
for Request in S do
    Find closest center to the request
    Move the associated server to the request
```

**Algorithm 2:** Known Zoning algorithm

**Theorem 3.4.** *The zoning algorithm has expected cost at most twice that of any online algorithm (up to a fixed additive cost) if the distribution is known. In particular,*

$$\mathbb{E}[\text{cost}(\textit{zoning})] \leq 2\,\mathbb{E}[\text{cost}(\mathbb{A})] + \Phi_0$$

*Proof.* At every time step we move the server associated to the closest center $k \in K^*$ (by the minimum bipartite matching) to the request point $x$. By the triangle inequality this distance is less than if we had moved the server to $k$ first, and then to $x$ after. Every request under such a modification therefore incurs at most two costs, movement from $k$ and movement

to $k$. The expected distance of any request to its closest center is $\int_{x \in X} d(x, K^*) P(\,\mathrm{d}x)$, so by linearity of expectation:

$$\mathbb{E}[\text{cost(zoning)}] \leq 2|S| \mathop{\mathbb{E}}_{x \in X}[d(x, K^*)] + d(\mathcal{C}_0, K^*) \overset{3.3}{\leq} 2\,\mathbb{E}[\text{cost}(\mathbb{A})] + \Phi_0$$

for requests $S$, any online algorithm $\mathbb{A}$, and where $\Phi_0 = d(\mathcal{C}_0, K^*)$ is an initial cost dependent only on the initial configuration and the distribution. □

## 3.3 Unknown Distribution

The most obvious next step is to deal with a model for which the sample distribution is not known. Perhaps the simplest approach would be to apply the zoning algorithm using centers calculated on some fixed number of initial observations. However, if a number of low probability points are observed, then it is possible that some of the $k$ centers do very little to lower the expected minimum distance. In other words, for some lower probability initial observations, the sample $k$-median centers may be extremely poor in comparison to those of the entire space.

An alternative approach is to calculate the new $k$-median centers on the entire set of samples seen up to a new request and use the zoning algorithm to serve it. Unfortunately computation of $k$-medians is computationally expensive, hence it would be preferable to amortize the time.

Hence our unknown zoning algorithm (line 3) uses $k$-median centers calculated on the observed samples, however we do so at geometrically spaced intervals. We will show that if $|S_0|$ is large enough then the expected cost of poorly served requests is not unbounded in the expected cost of an optimal algorithm.

Our analysis is heavily based on that given in [13], where it is shown that for large enough samples of a discrete sample set, the sample $k$-median is a good approximation of the $k$-median of the entire set, with high probability. Our analysis differs in numerous ways, including the fact that we are dealing with a more general probability spaces, and we are recomputing the $k$-median periodically over all historical samples, rather than just computing it once and assuming it would work over all future requests.

**Notation 3.5.** *We denote a partition of $X$ according to closest neighbours in $K^*$ by*

$$X = \coprod_{j=1}^{k} X_j^*$$

*where $X_j^* = \{\, x \in X : d(x, k_j^*) \leq d(x, k_i^*) \,\}$ where we break ties in any consistent manner.*

**Notation 3.6.** *We denote the $k$-median centers computed on a subset $S$ of the space $X$ by $K^S = \{k_j^S\}$. Note that this gives $K^X = K^*$.*

**Data:** Time scale factor $\gamma$, metric $d$, initial greedy count $|S_0|$, a sequence of requests $S = (x_1, \ldots, x_r)$, initial server positions
**Result:** Sequence of servers assigned in an online manner

```
Serve the first |S_0| requests greedily;
for i ← 1 to N do
    Compute α-approximate k-median centers, K, over all requests served so far;
    Associate every server with one of the centers;
    foreach Next γ^i |S_0| requests do
        Find closest center to request;
        Move the associated server to this center;
```

**Algorithm 3:** Unknown Zoning algorithm

**Notation 3.7.** *For a given sequence $S$, we then define $S^{(j)}$ to be the subsequence of points of $S$ that fall in $X_j^*$.*

**Lemma 3.8.** *For a given sequence $T$, $k$-median centers $K^*$ and some centers $K \subset X$, $|K| = k$, then for some fixed $j$: for any $x \in X_j^*$*

$$d(x, K) \leq d(x, K^*) + \frac{1}{|T^{(j)}|} \sum_{y \in T^{(j)}} [d(y, K^*) + d(y, K)]$$

*Proof.* Let $k_j^*$ denote the closest median to $x$, then by the triangle inequality

$$d(x, K) \leq d(x, k_j^*) + d(k_j^*, K) = d(x, K^*) + d(k_j^*, K)$$

Also by the triangle inequality, we have that for any $y \in T^{(j)}$

$$d(k_j^*, K) \leq d(k_j^*, y) + d(y, K) = d(y, K^*) + d(y, K)$$

This includes the minimizer and, using the fact that the arithmetic mean can be no smaller than the minimum,

$$d(k_j^*, K) \leq \min_{y \in T^{(j)}} [d(y, K^*) + d(y, K)] \leq \frac{1}{|T^{(j)}|} \sum_{y \in T^{(j)}} [d(y, K^*) + d(y, K)]$$

Combining the first and last gives the desired result. □

**Theorem 3.9.** *Suppose we service a sequence of requests $S$ sampled i.i.d in an online manner over $X$ using the unknown zoning algorithm. Given parameters: $\alpha$-approximate $k$-median algorithm, sequence factor $\gamma$, initial greedy run length $|S_0|$, and any desired $\epsilon > 0$, then if*

$$|S_0| \geq \frac{k\gamma^3(1+\alpha)^2}{m\epsilon^3} \max\left\{\frac{2080\Delta}{e\,\mathbb{E}[d(x, K^*)]\log\gamma}, 1024\epsilon\right\}$$

*where* $m = \min\{P(X_j^*) : P(X_j^*) > 0\}$, *then*

$$\mathbb{E}[\text{cost}(\textit{zoning})] \leq (2 + 2(1+\alpha)\gamma + \epsilon)\,\mathbb{E}[\text{cost}(\mathbb{A})] + \Delta|S_0|$$

*Proof.* We begin by splitting the sequence $S$ into subsequences $S = S_0 \frown S_1 \frown \cdots \frown S_N$ where the size increases as a geometric progression, i.e., $|S_{i+1}| = \gamma|S_i|$.

Without loss of generality we assume that the final subsequence $S_N$ also satisfies this relation, i.e., $|S_N| = \gamma|S_{N-1}|$. We may do this by appending a sequence of requests that coincide with the final server configuration, hence incurring no cost to this algorithm (as no server would need to move), yet it could potentially increase the cost of the optimal online algorithm.

We next define $T_i = S_0 \frown S_1 \frown \cdots \frown S_i$. The following properties will be used throughout:

1. $S_i = \gamma^i |S_0|$
2. $|T_i| = \frac{\gamma^{i+1}-1}{\gamma-1}|S_0|$
3. $\sum_{i=1}^{N} |T_{i-1}| = \sum_{i=1}^{N} \frac{\gamma^i - 1}{\gamma - 1}|S_0| \leq \frac{\gamma^{N+1}|S_0|}{(\gamma-1)^2} = |S|\frac{\gamma}{(\gamma-1)^2}$
4. For i.i.d requests $\mathbb{E}\,|S_i^{(j)}| = |S_i|P(X_j^*)$ and $\mathbb{E}\,|T_i^{(j)}| = |T_i|P(X_j^*)$.

There are three costs that need to be bounded: the greedy serving of $S_0$, the reconfiguration of the servers after new centers have been computed, and the serving of all requests in each $S_i$. The first and second of these are trivially bounded above by $\Delta|S_0|$ and $\Delta kN$, respectively, by assuming every server moves and incurs a maximum cost of $\Delta$, the diameter of the space.

In order to bound the expected serving cost of requests in $S_i$ for a fixed $1 \leq i \leq N$, we will first find an upper bound on the probability that the sample requests in each zone $X_j^*$ are "poor", in a certain sense. In particular, if

$$|S_i^{(j)}| \geq (1+\delta_1)\,\mathbb{E}\,|S_i^{(j)}| \quad \text{or if} \quad |T_{i-1}^{(j)}| \leq (1-\delta_2)\,\mathbb{E}\,|T_{i-1}^{(j)}|$$

The probability of any such event occuring when $P(X_j^*) \neq 0$ may be controlled by Chernoff bounds:

$$\mathbb{P}\left[\,|S_i^{(j)}| \geq (1+\delta_1)\,\mathbb{E}\,|S_i^{(j)}|\,\right] \leq \exp\left(-\tfrac{1}{3}\delta_1^2\,\mathbb{E}\,|S_i^{(j)}|\right) \leq \frac{3}{\delta_1^2\,\mathbb{E}\,|S_i^{(j)}|}$$

$$\mathbb{P}\left[\,|T_{i-1}^{(j)}| \leq (1-\delta_2)\,\mathbb{E}\,|T_{i-1}^{(j)}|\,\right] \leq \exp\left(-\tfrac{1}{2}\delta_2^2\,\mathbb{E}\,|T_{i-1}^{(j)}|\right) \leq \frac{2}{\delta_2^2\,\mathbb{E}\,|T_{i-1}^{(j)}|}$$

Define $m = \min\{\, P(X_j^*) : P(X_j^*) \neq 0\,\}$, then

$$\mathbb{E}\,|S_i^{(j)}| = |S_i|P(X_j^*) \geq |S_i|m$$
$$\mathbb{E}\,|T_{i-1}^{(j)}| = |T_{i-1}|P(X_j^*) \geq |T_{i-1}|m \geq |S_{i-1}|m = \tfrac{1}{\gamma}|S_i|m$$

By union bounds we obtain,

$$\begin{aligned}
&\mathbb{P}\left[\,\exists j : |S_i^{(j)}| \geq (1+\delta_1)\,\mathbb{E}\,|S_i^{(j)}| \text{ or } |T_{i-1}^{(j)}| \leq (1-\delta_2)\,\mathbb{E}\,|T_{i-1}^{(j)}| \text{ where } P(X_j^*) \neq 0\,\right] \\
&\leq \sum_{\substack{j=1,\\ P(X_j^*)\neq 0}}^{k} \left(\mathbb{P}\left[\,|S_i^{(j)}| \geq (1+\delta_1)\,\mathbb{E}\,|S_i^{(j)}|\,\right] + \mathbb{P}\left[\,|T_{i-1}^{(j)}| \leq (1-\delta_2)\,\mathbb{E}\,|T_{i-1}^{(j)}|\,\right]\right) \\
&\leq \sum_{\substack{j=1,\\ P(X_j^*)\neq 0}}^{k} \left(\frac{3}{\delta_1^2\,\mathbb{E}\,|S_i^{(j)}|} + \frac{2}{\delta_2^2\,\mathbb{E}\,|T_{i-1}^{(j)}|}\right) \leq k\left(\frac{3}{\delta_1^2|S_i|m} + \frac{2}{\delta_2^2\frac{1}{\gamma}|S_i|m}\right) = \frac{k}{|S_i|m}\left(\frac{3}{\delta_1^2} + \frac{2\gamma}{\delta_2^2}\right) \\
&= \frac{4k\gamma}{|S_i|m\delta_2^2}, \quad \text{where we define } \delta_1 \equiv \sqrt{\tfrac{3}{2\gamma}}\delta_2
\end{aligned}$$

As will be necessary momentarily, we will bound $\frac{|S_i^{(j)}|}{|T_{i-1}^{(j)}|}$ if no zones are "poor", for each $j$ such that $P(X_j^*) \neq 0$. Hence, by definition,

$$|S_i^{(j)}| < (1+\delta_1)\,\mathbb{E}\,|S_i^{(j)}| \quad \text{and} \quad |T_{i-1}^{(j)}| > (1-\delta_2)\,\mathbb{E}\,|T_{i-1}^{(j)}|,$$

giving

$$\begin{aligned}
\frac{|S_i^{(j)}|}{|T_{i-1}^{(j)}|} &< \frac{1+\delta_1}{1-\delta_2}\cdot\frac{\mathbb{E}\,|S_i^{(j)}|}{\mathbb{E}\,|T_{i-1}^{(j)}|} = \frac{1+\delta_1}{1-\delta_2}\cdot\frac{P(X_j^*)|S_i|}{P(X_j^*)|T_{i-1}|} \\
&< \frac{1+\delta_1}{1-\delta_2}\cdot\frac{|S_i|}{|S_{i-1}|} = \gamma\left(\frac{1+\delta_1}{1-\delta_2}\right) = \gamma\left(1+\frac{\delta_1+\delta_2}{1-\delta_2}\right) \\
&= \gamma\left(1+\left(1+\sqrt{\tfrac{3}{2\gamma}}\right)\cdot\frac{\delta_2}{1-\delta_2}\right) \\
&\leq \gamma\left(1+\left(1+\sqrt{\tfrac{3}{2\gamma}}\right)\cdot\underbrace{2\delta_2}_{0\leq\delta_2\leq 1}\right) \equiv \gamma\,(\,1+\hat{\epsilon}\,)
\end{aligned}$$

$$\hat{\epsilon} = 2\left(1+\sqrt{\tfrac{3}{2\gamma}}\right)\delta_2 \implies \delta_2^2 \equiv \frac{\hat{\epsilon}^2}{2^2\left(1+\sqrt{\frac{3}{2\gamma}}\right)^2} \geq \frac{\hat{\epsilon}^2}{16}, \quad \text{for } \gamma \geq 3/2$$

This then gives

$$\frac{4k\gamma}{|S_i|m\delta_2^2} \leq \frac{64k\gamma}{|S_i|m\hat{\epsilon}^2}$$

$$\mathbb{P}\left[\exists j : |S_i^{(j)}| \geq (1+\delta_1)\,\mathbb{E}\,|S_i^{(j)}| \text{ or } |T_{i-1}^{(j)}| \leq (1-\delta_2)\,\mathbb{E}\,|T_{i-1}^{(j)}| \text{ where } P(X_j^*) \neq 0\right]$$
$$\leq \min\left\{1, \frac{64k\gamma}{|S_i|m\hat{\epsilon}^2}\right\}$$

Note that in the algorithm the reconfiguration of a server only occurs if a request is associated to a computed center that is not nearest to its own associated server. The choice of reconfiguration bound, $\Delta kN$, permits us to assume that before serving requests in $S_i$ all servers are located at the computed approximate centers. This is an upper bound by the triangle inequality as movement to the center and then to a request is no larger than movement directly to that request.

Suppose we consider the cost of serving requests in $S_i$ for some fixed $1 \leq i \leq N$. Denote an $\alpha$-approximate $k$-median centers $K_{\text{app}}^{T_{i-1}}$ computed on $T_{i-1}$.

If a $j$ exists as given in the probability bounds, then we will bound the cost of all requests $S_i$ by $\Delta|S_i|$, otherwise for all $j$, $|T_{i-1}^{(j)}| > 0$ (from now on assume $P(X_j^*) \neq 0$, as we are dealing with expectation, so zero probability events are ignored).

Hence we may apply lemma 3.8, giving

$$2\sum_{x\in S_i} d(x, K_{\text{app}}^{T_{i-1}}) = 2\sum_{j=1}^{k}\sum_{x\in S_i^{(j)}} d(x, K_{\text{app}}^{T_{i-1}})$$
$$\overset{3.8}{\leq} 2\sum_{j=1}^{k}\sum_{x\in S_i^{(j)}}\left[d(x,K^*) + \frac{1}{|T_{i-1}^{(j)}|}\sum_{y\in T_{i-1}^{(j)}}\left(\,d(y,K^*) + d(y,K_{\text{app}}^{T_{i-1}})\,\right)\right]$$
$$= 2\sum_{x\in S_i} d(x,K^*) + 2\sum_{j=1}^{k}\frac{|S_i^{(j)}|}{|T_{i-1}^{(j)}|}\sum_{y\in T_{i-1}^{(j)}}\left(\,d(y,K^*) + d(y,K_{\text{app}}^{T_{i-1}})\,\right)$$
$$\leq 2\sum_{x\in S_i} d(x,K^*) + 2\gamma(1+\hat{\epsilon})\sum_{j=1}^{k}\sum_{y\in T_{i-1}^{(j)}}\left(d(y,K^*) + d(y,K_{\text{app}}^{T_{i-1}})\right)$$
$$= 2\sum_{x\in S_i} d(x,K^*) + 2\gamma(1+\hat{\epsilon})\sum_{y\in T_{i-1}} d(y,K^*) + 2\gamma(1+\hat{\epsilon})\sum_{y\in T_{i-1}} d(y,K_{app}^{T_{i-1}})$$

By definition of $\alpha$-approximate $k$-medians

$$\leq 2\sum_{x\in S_i} d(x,K^*) + 2\gamma(1+\hat{\epsilon})\sum_{y\in T_{i-1}} d(y,K^*) + 2\gamma(1+\hat{\epsilon})\alpha\cdot\sum_{y\in T_{i-1}} d(y,K^{T_{i-1}})$$

Given that $K^{T_{i-1}}$ is a $k$-median set of centers on $T_{i-1}$, then it must be less than the cost of using $K^*$.

$$\leq 2\sum_{x\in S_i} d(x,K^*) + 2\gamma(1+\alpha)(1+\hat{\epsilon})\sum_{y\in T_{i-1}} d(y,K^*)$$

Hence the expected cost of the zoning algorithm serving requests $S_i$, $1 \leq i \leq N$, is bounded above by the expected cost if all centers are "good" and the bound on probability of a "poor" center with the upper bound on cost. Hence the bound:

$$\mathbb{E}\left[2\sum_{x\in S_i} d(x,K^*) + 2(1+\alpha)\gamma(1+\hat{\epsilon})\sum_{y\in T_{i-1}} d(y,K^*)\right] + |S_i|\Delta\min\left\{1, \frac{64k\gamma}{|S_i|m\hat{\epsilon}^2}\right\}$$
$$\leq 2\sum_{x\in S_i}\mathbb{E}\left[d(x,K^*)\right] + 2(1+\alpha)\gamma(1+\hat{\epsilon})\sum_{y\in T_{i-1}}\mathbb{E}\left[d(y,K^*)\right] + \min\left\{\Delta|S_i|, \frac{64k\gamma\Delta}{m\hat{\epsilon}^2}\right\}$$

Now further suppose that $\frac{64k\gamma}{m\hat{\epsilon}^2} \leq |S_i|$, then

$$= 2|S_i|\,\mathbb{E}\left[d(x,K^*)\right] + 2(1+\alpha)\gamma(1+\hat{\epsilon})|T_{i-1}|\,\mathbb{E}\left[d(x,K^*)\right] + \frac{64k\gamma\Delta}{m\hat{\epsilon}^2}$$

Recalling that $|T_{i-1}| \leq |S_i|$,

$$\leq 2(1+(1+\alpha)(1+\hat{\epsilon})\gamma)|S_i|\,\mathbb{E}\left[d(x,K^*)\right] + \frac{64k\gamma\Delta}{m\hat{\epsilon}^2}$$

Hence if $\frac{64k\gamma}{m\hat{\epsilon}^2} \leq |S_0| < |S_1| < |S_2| < \cdots$, then the total expected cost, including reconfiguration and greedy is bounded above by

$$\mathbb{E}[\text{cost(zoning)}] \leq \Delta|S_0| + \Delta kN + \sum_{i=1}^{N}\left[2(1+(1+\alpha)(1+\hat{\epsilon})\gamma)|S_i|\,\mathbb{E}\left[d(x,K^*)\right] + \frac{64k\gamma\Delta}{m\hat{\epsilon}^2}\right]$$
$$= \Delta|S_0| + \Delta kN + 2(1+(1+\alpha)(1+\hat{\epsilon})\gamma)|S|\,\mathbb{E}\left[d(x,K^*)\right] + \frac{64k\gamma N\Delta}{m\hat{\epsilon}^2}$$

Recalling that $|S|\,\mathbb{E}[d(x,K^*)] \leq \mathbb{E}[\text{cost(opt)}]$,

$$\leq 2(1+(1+\alpha)(1+\hat{\epsilon})\gamma)\,\mathbb{E}[\text{cost(opt)}] + \Delta|S_0| + \Delta kN\left(1+\frac{64\gamma}{m\hat{\epsilon}^2}\right)$$
$$\leq 2(1+(1+\alpha)(1+\hat{\epsilon})\gamma)\,\mathbb{E}[\text{cost(opt)}] + \Delta|S_0| + \Delta kN\frac{65\gamma}{m\hat{\epsilon}^2}$$

Note that $N < \frac{\log(|S|/|S_0|)}{\log\gamma}$ (as $|S| > |S_N| = \gamma^N |S_0|$)

$$\leq 2(1+(1+\alpha)(1+\hat{\epsilon})\gamma)\,\mathbb{E}[\text{cost(opt)}] + \Delta|S_0| + \Delta k \frac{\log(|S|/|S_0|)}{\log\gamma}\frac{65\gamma}{m\hat{\epsilon}^2}$$
$$\leq 2(1+(1+\alpha)(1+\hat{\epsilon})\gamma)\,\mathbb{E}[\text{cost(opt)}] + \Delta|S_0| + |S|\,\mathbb{E}\left[d(x,K^*)\right]\frac{65\Delta k\gamma\log(|S|/|S_0|)}{|S|\,\mathbb{E}\left[d(x,K^*)\right]m\hat{\epsilon}^2\log\gamma}$$

Now given that $\frac{\log(x/|S_0|)}{x}$ is maximized at $x = |S_0|e$, with value $\frac{1}{|S_0|e}$, hence

$$\leq 2(1+(1+\alpha)(1+\hat{\epsilon})\gamma)\,\mathbb{E}[\text{cost(opt)}] + \Delta|S_0| + \mathbb{E}[\text{cost(opt)}]\frac{65\Delta k\gamma}{|S_0|e\,\mathbb{E}\left[d(x,K^*)\right]m\hat{\epsilon}^2\log\gamma}$$
$$= \left(2 + 2(1+\alpha)\gamma + 2(1+\alpha)\hat{\epsilon}\gamma + \frac{65\Delta k\gamma}{|S_0|e\,\mathbb{E}\left[d(x,K^*)\right]m\hat{\epsilon}^2\log\gamma}\right)\mathbb{E}[\text{cost(opt)}] + \Delta|S_0|$$

We now find a lower bound on $|S_0|$ in order to make the above multiplier at most $(2+2(1+\alpha)+\epsilon$.

If we choose $\hat{\epsilon}$ so that $\frac{\epsilon}{2} = 2(1+\alpha)\hat{\epsilon}\gamma$, then

$$\frac{1024k\gamma^3(1+\alpha)^2}{m\epsilon^2} = \frac{64k\gamma}{m\hat{\epsilon}^2} \leq |S_0|$$

and we want

$$\frac{65\Delta k\gamma}{|S_0|e\,\mathbb{E}[d(x,K^*)]m\hat{\epsilon}^2\log\gamma} \leq \frac{\epsilon}{2}$$

so

$$|S_0| \geq \frac{2}{\epsilon}\cdot\frac{65\Delta k\gamma}{e\,\mathbb{E}[d(x,K^*)]m\hat{\epsilon}^2\log\gamma} = \frac{2080\Delta k\gamma^3(1+\alpha)^2}{e\,\mathbb{E}[d(x,K^*)]m\epsilon^3\log\gamma}$$

Finally, if

$$|S_0| \geq \max\left\{\frac{2080\Delta k\gamma^3}{e\,\mathbb{E}[d(x,K^*)]m\epsilon^3\log\gamma}, \frac{1024k\gamma^3(1+\alpha)^2}{m\epsilon^2}\right\}$$
$$= \frac{k\gamma^3(1+\alpha)^2}{m\epsilon^3}\max\left\{\frac{2080\Delta}{e\,\mathbb{E}[d(x,K^*)]\log\gamma}, 1024\epsilon\right\}$$

then

$$\mathbb{E}[\text{cost(zoning)}] \leq (2+2(1+\alpha)\gamma+\epsilon)\,\mathbb{E}[\text{cost(opt)}] + \Delta|S_0|$$

□

# Chapter 4

# Lookahead Algorithm

In this chapter we give lookahead algorithms for the $k$-server problem, under the assumption of a known distribution. This algorithm perform on certain subsets of $\mathbb{R}^d$ using the $L_1$ and $L_2$ metrics. We provide experimental evidence supporting use of this algorithm.

## 4.1 The Algorithm

As mentioned in section 2.1 there are numerous algorithms that have been studied in relation to the $k$-server problem in both deterministic and randomized settings. The greedy algorithm which is known to have unbounded competitive ratio against an offline adversary offers some insight into lookahead improvements. It is not known whether the greedy algorithm is bounded competitive for unknown distribution models in general, though it is known to be optimal for the special case of uniform distribution with $k = 2$ on the unit circle [4].

Simple experimentation led to the observation that the density of the $k$ servers appears to be much higher towards the edges of the unit square when servers are chosen greedily. In other words, at any given request only a few servers are likely contenders.

**input :** Request $r$, Servers $\mathcal{P} = \{p_1, \ldots, p_k\}$
**output:** New servers $\hat{\mathcal{P}}$

**foreach** *Server* $p_i$ *in* $\mathcal{P}$ **do**
    $\mathcal{P}'_i \leftarrow \mathcal{P} \cup \{r\} \setminus \{p_i\}$
    Cost $C_i \leftarrow d(r, p_i)$
$\hat{\mathcal{P}} \leftarrow \mathcal{P}'_i$ for $i$ that minimizes $C_i$.

**Algorithm 4:** Greedy $k$-Server Algorithm Online Step

We recall that the work function algorithm essentially balances two unbounded algorithms

(retrospective and greedy). Our lookahead does much the same, where the retrospective cost is replaced by the expected minimum distance. The best configuration of servers in a stochastic setting awaiting a request would minimize the expected minimum distance. Hence some measure of deviation from optimal configuration (the $k$-median), helps guide the greedy choice of server.

**input :** Request $r$, Servers $\mathcal{P} = \{p_1, \ldots, p_k\}$, Parameter $\eta$
**output:** New servers $\hat{\mathcal{P}}$

**foreach** *Server* $p_i$ *in* $\mathcal{P}$ **do**
  $\mathcal{P}'_i \leftarrow \mathcal{P} \cup \{r\} \setminus \{p_i\}$
  Cost $C_i \leftarrow d(r, p_i) + \eta\, \mathbb{D}(\mathcal{P}'_i)$
$\hat{\mathcal{P}} \leftarrow \mathcal{P}'_i$ for $i$ that minimizes $C_i$.

**Algorithm 5:** Lookahead $k$-Server Algorithm Online Step

where $\mathbb{D}(\mathcal{P}'_i)$ denotes the expected minimum distance from servers in $\mathcal{P}'_i$ to points in some probabilistic metric space.

## 4.2 Background

### 4.2.1 Generalized Voronoi Diagrams

The topic of Voronoi diagrams has a long history, having first been studied, at least informally, by Descartes [14]. They have had numerous applications throughout the centuries, from the first supporting evidence of John Snow's theory as to the source of a cholera outbreak in the 1850s [32] through to numerous applications in modern computer graphics, e.g., [29].

The core idea behind Voronoi diagrams is, given a set of points, to partition a space into regions corresponding to each point such that a region is most similar to its point. A concrete example for which they were initially studied was to partition the Euclidean plane into closest regions of a point set. More generally,

**Definition 4.1.** *Given a metric space $(X, d)$, a set of distinct* ***sites*** $\mathcal{P} = \{p_1, \ldots, p_n\} \subseteq X$ *induce a set of* ***cells*** $\mathcal{R} = (R_i)_1^n$ *such that*

$$R_i = \{x \in X \mid d(x, p_i) \leq d(x, p_j) \text{ for all } j = 1, \ldots, n\}$$

*The* ***Voronoi diagram*** $\mathrm{Vor}(\mathcal{P})$ *is then the pair* $(\mathcal{P}, \mathcal{R})$.

The simplest approach to calculate general Voronoi diagrams is to consider the intersection of halfplanes, which will be explained shortly. This is generally a very poor approach as

it requires pairwise comparisons of all sites. A well-known algorithm that efficiently calculates the Voronoi diagram on $\mathbb{R}^2$ with the usual $L_2$-norm is given by Fortune's algorithm, a sweepline algorithm that runs in $O(n \log n)$ time using $O(n)$ storage [17]. Fortune's algorithm relies on the geometric properties of parabolae, and hence cannot be extended easily to general metrics. Sweepline methods are also known for $L_1$ and $L_\infty$ [40] and more general algorithms are known for more general norm-induced metrics [12], though they are based on dynamic programming.

**Definition 4.2.** *The **closed halfplane** of a site $p_i$ with respect to $p_j$, $i \neq j$, is given by*

$$h_{p_i}(p_j) = \{x \in X \mid d(x, p_i) \leq d(x, p_j)\}$$

It is clear from this definition and that of the Voronoi diagram, that

$$R_i = \bigcap_{j \neq i} h_{p_i}(p_j) \tag{4.1}$$

**Theorem 4.3.** *If $M \subset \mathbb{R}^d$ is a convex polytope then for a metric spaces induced by $\|\cdot\|_1$, $\|\cdot\|_2$ or $\|\cdot\|_\infty$, all Voronoi regions are star-shaped polytopes with respect to its corresponding site.*

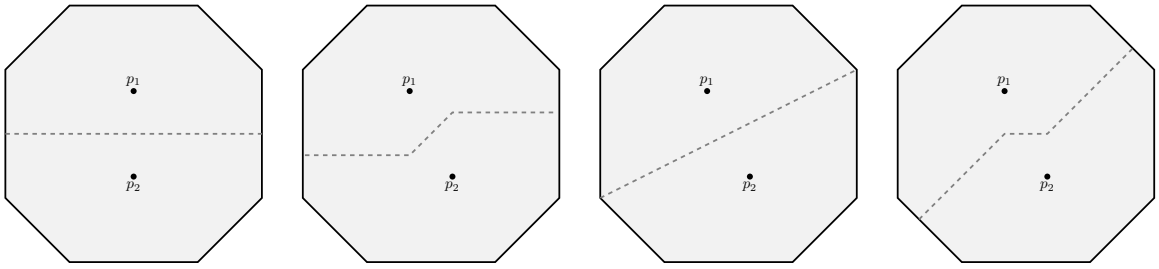



Figure 4.1: Bisectors through an octogonal subspace $X \subset \mathbb{R}^2$, (a) A "trivial" bisector that is identical for any $1 \leq p \leq \infty$, (b) $L_1$ metric, (c) $L_2$ metric, (d) $L_\infty$ metric.

**Definition 4.4.** *A $d$-**simplex** $S$ is a $d$-dimensional polytope defined by $d+1$ affinely independent vertices, i.e.,*

$$S = \left\{ \sum_{i=0}^{d} \lambda_i u_i \mid \lambda_i \geq 0, \sum_{i=0}^{d} \lambda_i = 1 \right\}$$

*where $u_1 - u_0, \ldots, u_d - u_0$ are linearly independent vectors. If such vectors are linearly dependent then the simplex is said to be degenerate.*

**Remark 4.5.** *A line segment is a 1-simplex and a triangle is a 2-simplex. Furthermore, note that simplex is degenerate if and only if its $d$-hypervolume is zero in $\mathbb{R}^d$.*

**Theorem 4.6.** *For any $(X, L_p)$ with $p = 1, 2, \infty$ as in theorem 4.3, we may form a* ***simplicial decomposition*** $\mathbb{S}_{(X,L_p)}(p) = \mathbb{S}(p)$ *of every Voronoi cell.*

*This means that there exist simplices $\{S\} = \mathbb{S}(p)$ with $u_0 = p$ for each $S$, such that*

$$\bigcup_{S\in\mathbb{S}(p)} \operatorname{int}(S) \subset R = \bigcup_{S\in\mathbb{S}(p)} S$$

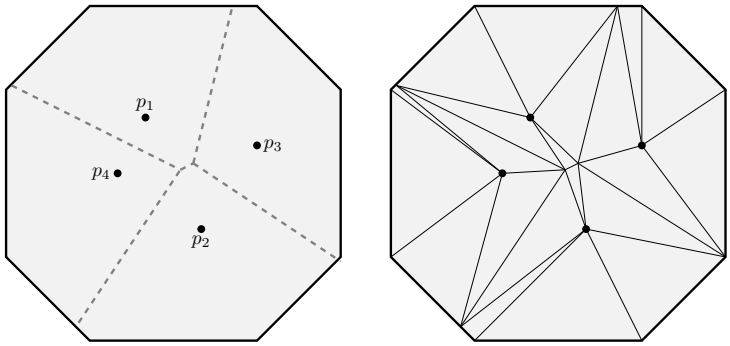



Figure 4.2: (a) Voronoi diagram using $L_2$ metric, (b) Simplicial decomposition using sites.

## 4.3 Expected Minimum Distance to a Set of Points

### 4.3.1 General Setting

For a Voronoi diagram $\operatorname{Vor}(\mathcal{P})$ with zero measure boundary, the **expected minimum distance of points**, $\mathbb{D}_{(X,d)}(\mathcal{P}) = \mathbb{D}(\mathcal{P})$, is given by

$$\mathbb{D}(\mathcal{P}) = \int_X \min_{p\in\mathcal{P}} d(\mathbf{r}, p) P(\mathrm{d}\mathbf{r}) = \sum_{(p,R)\in\operatorname{Vor}(\mathcal{P})} \int_R d(\mathbf{r}, p) P(\mathrm{d}\mathbf{r})$$

For spaces with a simplicial decomposition $\mathbb{S}(p)$ for each $R \in \mathcal{R}$, we may use the decomposition to give the following

$$\mathbb{D}(\mathcal{P}) = \sum_{(p,R)\in\operatorname{Vor}(\mathcal{P})} \int_R d(\mathbf{r}, p) P(\mathrm{d}\mathbf{r}) = \sum_{(p,R)\in\operatorname{Vor}(\mathcal{P})} \sum_{S\in\mathbb{S}(p)} \int_S d(\mathbf{r}, p) P(\mathrm{d}\mathbf{r})$$

In the special case of a uniform distribution, the probability distribution $S$ is simply given by $P(\,\mathrm{d}\mathbf{r}) = \frac{\mathrm{d}\mathbf{r}}{V(X)}$, where $V(\cdot)$ denotes the hyper-volume. This then gives

$$\mathbb{D}(\mathcal{P}) = \frac{1}{V(X)} \sum_{(p,R)\in\operatorname{Vor}(\mathcal{P})} \sum_{S\in\mathbb{S}(p)} \int_S d(\mathbf{r}, p)\,\mathrm{d}\mathbf{r}$$

For convenience we will refer to the integral $\int_S d(\mathbf{r}, p)\,\mathrm{d}\mathbf{r}$ simply as the **distance integral**.

### 4.3.2 Monte-Carlo Method

A simple approximation of the distance integral may also be computed using Monte-Carlo sampling.

**input :** $\mathcal{P} = \{p_1, \ldots, p_k\}$,
$r$ the number of samples.
**output:** Approximation of $\mathbb{D}(\mathcal{P})$

```
1 Compute Voronoi diagram with respect to P.
2 total_cost = 0
3 for nsamp = 1 to r do
4 |   Sample some point x from the space.
5 |   Find which cell R contains x.
6 |   Add the distance from x to the center p of R.
7 Approximation is total_cost /r.
```

**Algorithm 6:** Sampling Approximation

In the case of $X = [0, 1]^2$, the corresponding Voronoi diagrams of $L_1$ and $L_2$ have vertices that grow linearly with number of sites (which is $k$ in this case). Using trapezoidal maps we can then perform closest site lookup in expected $O(\log k)$ (e.g., [10]).

In practice the Monte-Carlo approach is slow to converge, taking an impractical amount of samples to be accurate to 3 decimal places.

### 4.3.3 $L_2$ on subsets of $\mathbb{R}^2$ uniformly distributed

We present a simple form for the Euclidean distance integral over a simplex in $\mathbb{R}^2$. The techniques appear to be too cumbersome for the, perhaps futile task of finding a simple form for $\mathbb{R}^d$ with $L_2$. Hence we shall only consider $\mathbb{R}^2$.

As previously mentioned in remark 4.5, if we consider $\mathbb{R}^2$ then all simplicies are triangles. For any triangle $S$ we may form a closed-form expression for the distance integral $\int_S d(\mathbf{r}, p)\, \mathrm{d}\mathbf{r}$. This will allow for us to calculate an exact value for the expected minimum distance of points uniformly distributed on $M$ to points in $\mathcal{P}$.

For any degenerate triangle, i.e., a triangle with area 0, the integral is clearly 0 as we are integrating over a set of measure 0. Considering non-degenerate triangles instead allows us to first show a closed-form expression on a right-angled triangle which we will then extend to general triangles.

## For Right-Angled Triangles

**Theorem 4.7.** *For a right-angled triangle $S$ (as labelled in fig. 4.3), then the distance integral has the closed-form:*

$$\int_S d(\mathbf{r}, u_0)\,\mathrm{d}\mathbf{r} = \frac{\alpha\beta\gamma}{6} + \frac{\beta^3}{6}\log\left(\frac{\alpha+\gamma}{\beta}\right)$$

*where $\alpha = \|u_2 - u_1\|_2$, $\beta = \|u_1 - u_0\|_2$ and $\gamma = \|u_2 - u_0\|_2$.*

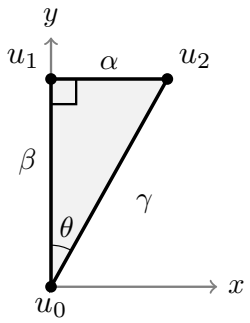



Figure 4.3: Generic right-angled triangle.

*Proof.* The Euclidean distance is rotationally-invariant and translation-invariant, so we may assume $u_0$ is at the origin and the orientation is the same as fig. 4.3. We then compute

$$\begin{aligned}
\int_S d(\mathbf{r}, u_0)\,\mathrm{d}\mathbf{r} &= \int_0^\beta \int_0^{y\tan\theta} \sqrt{x^2+y^2}\,\mathrm{d}x\,\mathrm{d}y \\
&= \int_0^\beta \frac{1}{2}\left[x\sqrt{x^2+y^2} + y^2\log\left(x+\sqrt{x^2+y^2}\right)\right]_0^{y\tan\theta}\,\mathrm{d}y \\
&= \frac{1}{2}\int_0^\beta y^2\tan\theta\sec\theta + y^2\log\left(\tan\theta+\sec\theta\right)\,\mathrm{d}y \\
&\overset{(*)}{=} \frac{\beta^3}{6}\left[\frac{\sqrt{1-(\beta/\gamma)^2}}{(\beta/\gamma)}\cdot\frac{1}{(\beta/\gamma)} + \log\left(\frac{\sqrt{1-(\beta/\gamma)^2}}{(\beta/\gamma)} + \frac{1}{(\beta/\gamma)}\right)\right] \\
&= \frac{\beta\gamma}{6}\sqrt{\gamma^2-\beta^2} + \frac{\beta^3}{6}\log\left(\frac{\gamma+\sqrt{\gamma^2-\beta^2}}{\beta}\right) \\
&= \frac{\alpha\beta\gamma}{6} + \frac{\beta^3}{6}\log\left(\frac{\alpha+\gamma}{\beta}\right)
\end{aligned}$$

where $(*)$ follows because $\theta = \cos^{-1}(\beta/\gamma)$. □

Here we reprove a result from [11], a simple corollary of our theorem. The proof we give motivates an approach to computing the distance integral over general triangles.

**Corollary 4.8.** *The expected distance from a fixed vertex to points in an equilateral triangle with unit side length is given by*

$$\frac{1}{12}(4 + 3\log 3)$$

*Proof.* If we bisect an equilateral triangle on the fixed vertex then we obtain two right-angled triangles. These two identical right-angled triangles have side lengths $\gamma = 1$, $\alpha = 1/2$ and $\beta = \sqrt{3}/2$, corresponding to fig. 4.3. Hence the distance integral over the equilateral is merely the sum of it over the right-angled triangles.

The expected distance is given by normalizing by the area of an equilateral triangle, which is $\frac{\sqrt{3}}{4}\gamma^2$. Therefore the expected distance is given by

$$\begin{aligned}
\frac{2}{V(S_{\text{eq}})}\int_{S_{\text{right}}} d(\mathbf{r}, u_0)\,\mathrm{d}\mathbf{r} &= \frac{2}{\frac{\sqrt{3}}{4}\gamma^2}\left[\frac{\alpha\beta\gamma}{6} + \frac{\beta^3}{6}\log\left(\frac{\alpha+\gamma}{\beta}\right)\right] \\
&= \frac{2}{\frac{\sqrt{3}}{4}(1)^2}\left[\frac{(\frac{1}{2})(\frac{\sqrt{3}}{2})(1)}{6} + \frac{(\frac{\sqrt{3}}{2})^3}{6}\log\left(\frac{(\frac{1}{2})+(1)}{(\frac{\sqrt{3}}{2})}\right)\right] \\
&= \frac{1}{6}(2 + 3\log\sqrt{3}) = \frac{1}{12}(4 + 3\log 3)
\end{aligned}$$

□

## For General Triangles

We now show how to decompose any non-degenerate triangle into a form so that we may use theorem 4.7. In order to do this we parameterize the line segment between $u_1$ and $u_2$, giving $w_t = (1-t)u_1 + tu_2$.

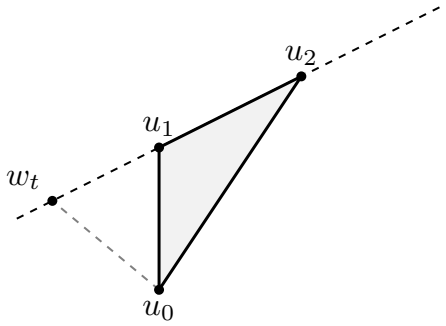



Figure 4.4: Extension of line segment $\overline{u_1u_2}$.

Our intention is to use $w_t$ to form right-angled triangles. To do this we need to find the

value of $t = t^*$ for which $\overline{w_t u_0}$ and $\overline{w_t u_1}$ are perpendicular.

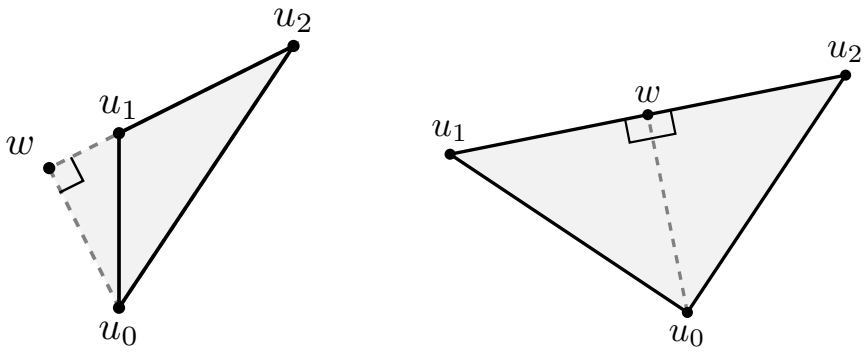
$$0 = \langle w_t - u_0, w_t - u_1 \rangle = \langle t u_2 + (1-t)u_1 - u_0, t(u_2 - u_1) \rangle$$

Solving for the non-trivial solution,

$$t^* = \frac{\langle u_2 - u_1, u_1 - u_0 \rangle}{\|u_2 - u_1\|^2} \tag{4.2}$$

Denote the solution position $w \equiv w_{t^*}$. There are then three distinct cases:

1. $t^* = 0$ or $t^* = 1$, then the triangle is right-angled.
2. $t^* \notin [0, 1]$, then we may calculate the distance integral over $u_0 u_1 u_2$ by computing the absolute difference between the distance integral over $u_0 w u_1$ and $u_0 w u_2$. We use the absolute difference so that we need not consider whether $t^* < 0$ or $t^* > 1$. This may be seen in fig. 4.5a.
3. $t \in (0, 1)$, then we may calculate the distance integral over $u_0 u_1 u_2$ by computing the addition of the integral over $u_0 w u_1$ and $u_0 w u_2$, as seen in fig. 4.5b.




(a) $\mathbf{t^* \notin [0, 1]}$. $\overline{wu_0}$ forms two new right-angled triangles, $u_0 w u_1$ and $u_0 w u_2$.

(b) $\mathbf{t^* \in (0, 1)}$. $\overline{wu_0}$ splits $u_0 u_1 u_2$ into two right-angled triangles, $u_0 w u_1$ and $u_0 w u_2$.

Figure 4.5: The two distinct cases of triangles represented in terms of right-angled triangles.

### 4.3.4 $L_1$ on subsets of $\mathbb{R}^d$ uniformly distributed

It is known that the volume of $d$-simplex with vertices $\{u_i\}$ for $i = 0, 1, \ldots, d$, satisfies:

$$d! V(S) = |\det(u_1 - u_0, u_2 - u_0, \ldots, u_d - u_0)|$$

In order to get a nice expression for the distance integral, we restrict ourselves to simplices that fall within a fixed orthant. To apply this in general, we may form a new simplicial decomposition by splitting every simplex $S \ni p_i$ into further simplices by intersecting planes parallel to the orthant axes, where each contains $u_0$.

**Theorem 4.9.** *The distance integral over any simplex $S \subset \mathcal{O}$, where $\mathcal{O}$ is an orthant, has explicit value*

$$\int_S d(\mathbf{r}, u_0)\,\mathrm{d}\mathbf{r} = \frac{V(S)}{d+1}\sum_{i=1}^{d}\|u_i - u_0\|_1 = \frac{|\det(u_1 - u_0, \ldots, u_d - u_0)|}{(d+1)!}\sum_{i=1}^{d}\|u_i - u_0\|_1$$

*where the simplex is defined with vertices $u_0, \ldots, u_d$.*

*Proof.* General Barycentric coordinates are given by

$$\mathbf{r} = \lambda_1 u_1 + \cdots + \lambda_n u_d + (1 - \lambda_1 - \cdots - \lambda_d)u_0,$$

where $0 \le \lambda_i$ and $\sum_{i=1}^{d}\lambda \le 1$.

Let $\sigma : \{1, \ldots, d\} \to \{1, \ldots, d\}$ be a permutation, then define

$$\int^{[d],\sigma} \equiv \int_0^1 \int_0^{1-\lambda_{\sigma(d)}} \int_0^{1-\lambda_{\sigma(d)}-\lambda_{\sigma(d-1)}} \cdots \int_0^{1-\sum_{i=2}^{d}\lambda_{\sigma(i)}}$$

then integration over an $d$-simplex $S$ using Barycentric coordinates is given by

$$\int_S d(\mathbf{r}, u_0)\,\mathrm{d}\mathbf{r} = d!V(S)\int^{[d],\sigma} f(\lambda_{\sigma(1)}u_1 + \cdots + \lambda_{\sigma(d)}u_d + (1 - \lambda_{\sigma(1)} - \cdots - \lambda_{\sigma(d)})u_0)\,\mathrm{d}\lambda_{\sigma(1)}\cdots\mathrm{d}\lambda_{\sigma(d)}$$

As the $L_1$ metric is translation-invariant, we may calculate assuming $u_0 = (0, \ldots, 0)$ using the mapping $u_i - u_0 \mapsto u_i'$. Recalling the result given in theorem C.2, we use the fact that every point is in the same orthant

$$\begin{aligned}
\int_S d(\mathbf{r}, u_0)\,\mathrm{d}\mathbf{r} &= \int_S \|\mathbf{r} - u_0\|_1\,\mathrm{d}\mathbf{r} \\
&= d!V(S)\int^{[d],\sigma} \|\lambda_{\sigma(1)}u_1 + \cdots + \lambda_{\sigma(d)}u_d\|_1\,\mathrm{d}\lambda_{\sigma(1)}\cdots\mathrm{d}\lambda_{\sigma(d)} \\
&\overset{C.2}{=} d!V(S)\int^{[d],\sigma} \left(\lambda_{\sigma(1)}\|u_1\|_1 + \cdots + \lambda_{\sigma(d)}\|u_d\|_1\right)\mathrm{d}\lambda_{\sigma(1)}\cdots\mathrm{d}\lambda_{\sigma(d)}
\end{aligned}$$

As the integral is identical under permutation $\sigma$, the resulting linear polynomial in $\|u_1\|_1, \ldots, \|u_d\|_1$ has identical coefficients, so

$$
\begin{aligned}
&= d!V(S) \sum_{i=1}^{d} \|u_i\|_1 \int^{[d]} \lambda_1 \, \mathrm{d}\lambda_1 \cdots \lambda_d \\
&= d!V(S) \sum_{i=1}^{d} \|u_i\|_1 \int^{[d-1]} \int_0^{1-\sum_{i=2}^{d} \lambda_i} \lambda_1 \, \mathrm{d}\lambda_1 \cdots \lambda_d \\
&= d!V(S) \sum_{i=1}^{d} \|u_i\|_1 \int^{[d-2]} \int_0^{1-\sum_{i=2}^{d-1} \lambda_i} \frac{1}{2} \left(1 - \sum_{i=2}^{d-1} \lambda_i\right)^2 \mathrm{d}\lambda_2 \cdots \lambda_d \\
&= d!V(S) \sum_{i=1}^{d} \|u_i\|_1 \int^{[d-2]} \frac{1}{2 \cdot 3} \left(1 - \sum_{i=2}^{d-2} \lambda_i\right)^3 \mathrm{d}\lambda_3 \cdots \lambda_d \\
&= \cdots \\
&= d!V(S) \sum_{i=1}^{d} \|u_i\|_1 \frac{1}{2 \cdot \cdots \cdot d \cdot (d+1)} \\
&= \frac{V(S)}{d+1} \sum_{i=1}^{d} \|u_i\|_1 = \frac{|\det(u_1, \ldots, u_d)|}{(d+1)!} \sum_{i=1}^{d} \|u_i\|_1
\end{aligned}
$$

□

**Remark 4.10.** *We have provided a few methods for calculating the expected minimum distance for a given set of points. Recall the definition of the k-medians (definition 3.1):*

$$K^* = \operatorname*{arg\,min}_{K \subset X, |K|=k} \mathbb{E}[d(\mathbf{r}, K)] = \operatorname*{arg\,min}_{K \subset X, |K|=k} \mathbb{D}(K)$$

*Hence the k-medians is a configuration of Voronoi sites that minimizes the expected minimum distance from a point sampled in the space to the nearest site.*

## 4.4 Experiments and Results

We implemented a simplicial decomposition of Voronoi diagram, allowing for computation of expected minimum distance to servers with $L_2$, using section 4.3.3. This was written in Python 2.7.

We then calculated the ratio of total costs of lookahead to greedy, presenting the mean across repetitions fig. 4.7. Note that at $\eta = 0$ the lookahead is equivalent to greedy, and hence should always be 1.

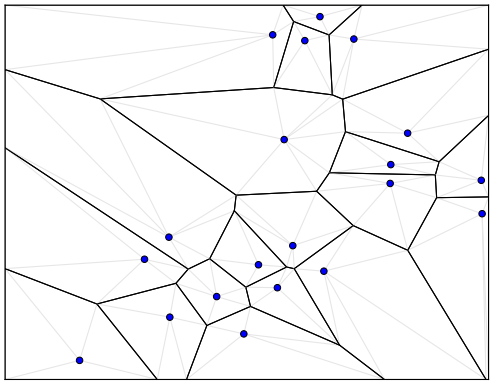


Figure 4.6: A simplicial decomposition with bolded Voronoi diagram produced in implementation, where $k = 20$.

Though no clear pattern for values of $\eta$ emerge, the lookahead algorithm appears to outperform greedy on average by $1 - 3\%$ for a range of values of $\eta$. This offers some empirical evidence that greedy is not optimal for general values of $k$.

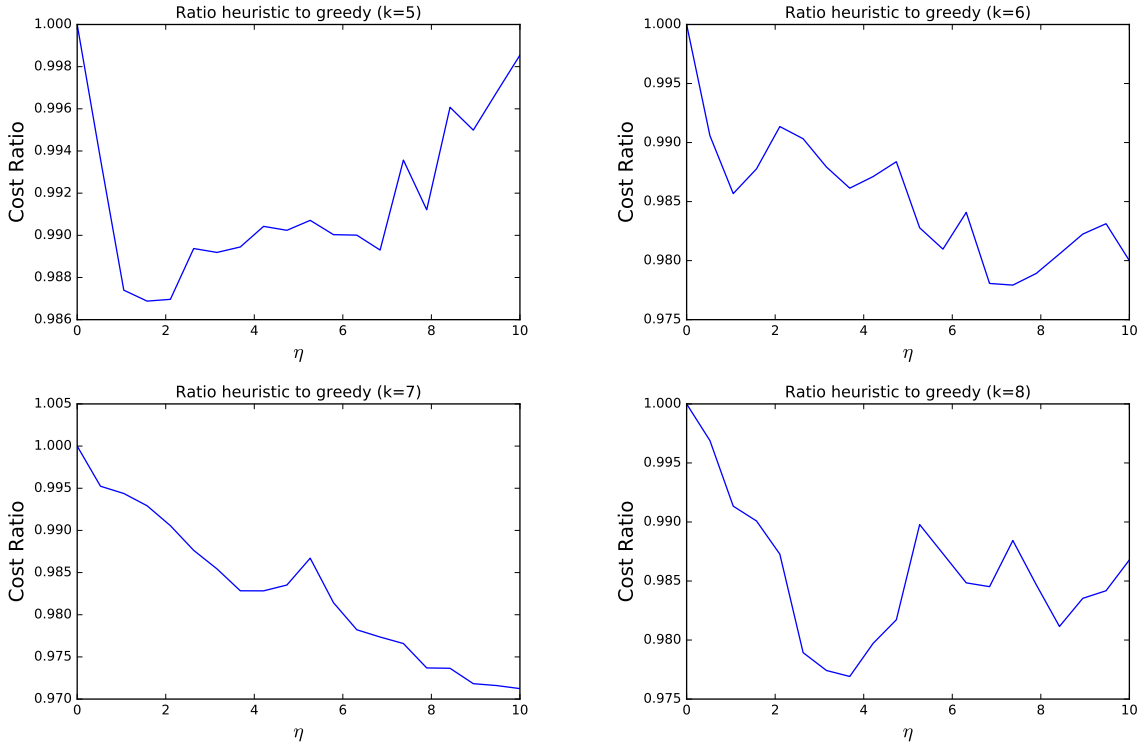



Figure 4.7: Graphs showing the mean ratio of costs in serving 300 requests, averaged across 10 repetitions, for various different values of $\eta$.

It is worth noting that this is merely a presentation of mean results. There were instances where greedy out performed the lookahead. Another point of note is that computation of an accurate expected minimum distance seems important to the benefit of this lookahead: when implemented using a Monte-Carlo method with $10,000$ samples there was no noticeable difference to greedy. Upon further exploration the Monte-Carlo method rarely produced approximations of expected minimum distance accurate to more than a few decimal places, and the error dominates the benefit of inclusion in the lookahead step.

**Remark 4.11.** *Although in our implementation we computed Voronoi diagrams at every time step, it is possible to do insertion and deletion in $O(k)$ [29], which would be ideal for such an online algorithm.*

# Chapter 5

# Stacker-Crane Approximations

In this chapter we discuss some algorithms that approximately solve the Stacker-Crane Problem.

## 5.1 Introduction

Here we shall consider two new algorithms for ESCP (definition 2.7), the **Approximate SPLICE Algorithm (ASPLICE)** and the **Probabilistically Locally Generated Algorithm (PLG)**.

We will show that ASPLICE is asymptotically near optimal almost surely for distributions with bounded p.d.fs. Under a further restriction that $P_p = P_d$, PLG is a linear-time asymptotically almost-surely near-optimal algorithm.

The key idea of PLG is that, due to the fact that $\phi_P = \phi_D$, if the request size $n$ is large enough then every pickup should be "close" to a delivery point. Using a probabilistic partition, we may greedily form a tour by connecting local points with respect to the partition and non-local connections once local points have been exhausted. We show that non-local connections are "rare" and that our approximation may be made arbitrarily good (in limit) by refining the partition.

The idea behind ASPLICE is that if we partition the space in a certain way then by using the Transportation Problem to connect regions of points, we approximate the SPLICE algorithm.

## 5.2 Background and Supporting Theorems

### 5.2.1 Probabilistic Hyperoctrees

We begin by defining the **probabilistic hyperoctree**, a spatial partitioning data structure essential to our algorithm. Both probabilistic quadtrees and hyperoctrees have been extensively studied in literature, where each is an independent generalization of a quadtree.

Throughout this section we assume the points are distributed over a compact space $\Omega \subset \mathbb{R}^d$ which, without loss of generality, we may assume come from the unit hypercube $\Omega = [0,1]^d$. This is possible as any compact space in $\mathbb{R}^d$ is bounded and therefore may be covered by some translation and scaling of the unit hypercube, giving a new distribution $\hat{f}(\mathbf{r})$ where $\hat{f}(\mathbf{r}) = f(\alpha\mathbf{r})$ on the usual domain and $\hat{f} = 0$ on the extension. As we use the Euclidean metric, it follows that we may just use the unit hypercube with probabilities scale and points mapped accordingly.

**Definition 5.1.** *A **hyperoctree** is a space-partitioning tree on $[0,1]^d$ with the property that every vertex has a cell of space and every vertex either represents a leaf or it divides its cell into $2^d$ children representing each region of an orthant centered in the middle of the cell. The cells corresponding to the leaf nodes are denoted $\mathcal{C}_i$ where $i = 1, \ldots, N$, with side lengths $\ell_i$.*

The most common types of hyperoctrees are the quadtree and octree, where $d = 2$ and 3 respectively, however the results here are proven in the general setting for the sake of completeness.

**Definition 5.2.** *Given some fixed probability $\hat{p} \in (0,1)$, a **probabilistic hyperoctree** $\mathcal{P}(\hat{p})$ has the property that a cell is divided if the probability measure of the cell is not less than $\hat{p}$. We denote the corresponding probability measure of a cell $\mathcal{C}$ by $P(\mathcal{C})$.*

*We refer to the set of leaf cells as the **PHT decomposition**.*

**Remark 5.3.** *It is not always possible to construct a finite PHT decomposition for any $\hat{p}$, e.g., the Dirac measure on $[0,1]^d$ cannot be decomposed with $\hat{p} = \frac{1}{2}$.*

*Fortunately, as we shall show, absolute continuity suffices for a finite PHT to exist, but we also consider a further restriction to gain some control over some properties of the PHT decomposition.*

**Theorem 5.4.** *If the probability measure is absolutely continuous, then a PHT may be constructed.*

*Proof.* Given any $\hat{p}$ there is some positive number $\delta > 0$ such that $P(A) < \hat{p}$ for all Borel sets $A$ of Lebesgue measure less than $\delta$. This follows from the equivalence of absolute continuity on finite measures over $\mathbb{R}^d$.

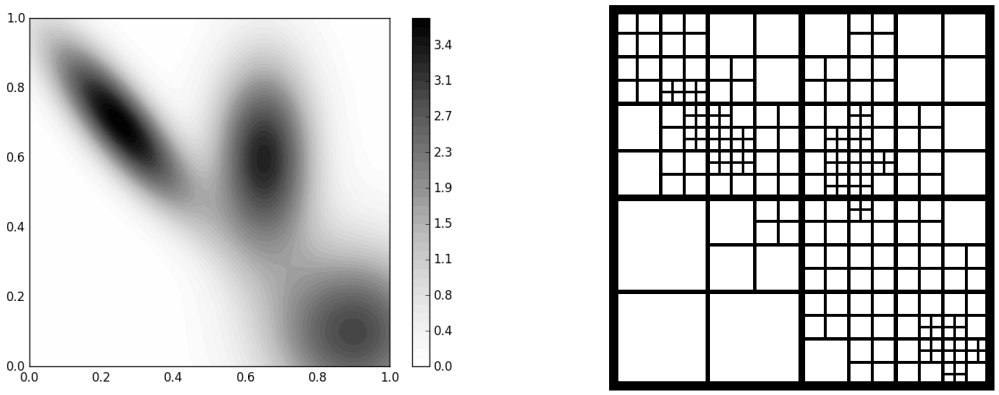



Figure 5.1: (a) Example Gaussian Mixture Model, (b) Corresponding PHT decomposition with $\hat{p} = 1\%$.

As the PHT only ever constructs hypercubes, which are Borel sets, the smallest side length is at least $\hat{\ell} \equiv \sqrt[d]{\delta}$. □

**Remark 5.5.** *The lower bound on smallest side length $\hat{\ell}$ allows a simple bound on $N$, the number of leaf cells. Suppose the entire hypercube were split into cubes with side length $\hat{\ell}$, then clearly this partition is a finer partition of PHT, hence*

$$\frac{1}{\delta} < \left(\frac{1}{\hat{\ell}}\right)^d < N < \left(\frac{1}{\hat{\ell}/2}\right)^d < \frac{2^d}{\delta} \tag{5.1}$$

**Theorem 5.6.** *If the p.d.f, $\phi$, is bounded on $[0,1]^d$ by $C$, then for any $\epsilon > 0$ we may find a $\hat{p}$ with corresponding PHT so that*

$$\sqrt{d}\sum_{j=1}^{N} P(\mathcal{C}_j)\ell_j < \epsilon$$

*Proof.* By absolute continuity there exists some $m$ for which $C/2^{md} \leq \hat{p} < 2^d C/2^{md}$ implies $P(\mathcal{C}) < \hat{p}$ for all cells in the PHT decomposition. The finest grid to this level has $\hat{N} = 2^{md}$ hypercubes. Then,

$$\frac{\hat{N}}{C} \geq \frac{1}{\hat{p}} > \frac{\hat{N}}{2^d C} > \frac{N}{2^d C} \implies N < \frac{2^d C}{\hat{p}}$$

Next, as the combined volume of all cell must total to 1, applying Hölder's inequality:

$$\sqrt{d}\sum_{j=1}^{N} P(\mathcal{C}_j)\ell_j \leq \sqrt{d}\left(\sum_{j=1}^{N} P(\mathcal{C}_j)^{d/(d-1)}\right)^{1-1/d} \underbrace{\left(\sum_{j=1}^{N} \ell_j^d\right)^{1/d}}_{=1}$$

$$\leq \sqrt{d} N^{1-1/d}\hat{p} \leq \sqrt{d} 2^{d-1} C^{1-1/d}\hat{p}^{1/d}$$

Hence we can ensure the bound for $\epsilon > 0$ by choosing $\hat{p}$ so that

$$\hat{p} < \frac{\epsilon^d}{2^{d(d-1)} C^{d-1} d^{d/2}}$$

□

**Definition 5.7.** *If the PHT is constructed on a known distribution then we refer to it as the* ***true PHT****, whereas if it is constructed from a sample set $S$, where $P(\mathcal{C}) = |\mathcal{C} \cap S|/|S|$, then we refer to it as the* ***sample PHT****.*

**Theorem 5.8.** *The sample PHT and true PHT are identical almost always.*

*Proof.* Take any leaf $\mathcal{C} \in \mathcal{P}(\hat{p})$ and its parent vertex $\mathcal{C}'$, then the stopping condition on cell division is equivalent to

$$P(\mathcal{C}) < \hat{p} \leq P(\mathcal{C}')$$

Now when we form a sample PHT it may be compared to the true PHT by considering the leaf nodes. If the two PHTs differ then, for at least one leaf cell $\mathcal{C}$ in the true distribution, either $\frac{|\mathcal{C} \cap S|}{|S|} > \hat{p}$, i.e., the cell was split in the sample PHT so it is not a leaf of the sample PHT, or $\frac{|\mathcal{C}' \cap S|}{|S|} < \hat{p}$, i.e., the parent cell (which may not even be present in the sample PHT) would not have been partitioned, hence $\mathcal{C}$ would not be in the sample PHT.

We will show that an even larger event set happens a finite number of times, almost surely.

Whether or not a point in the sample falls in cell $\mathcal{C}$ is modeled by a Bernoulli random variable with probability $P(\mathcal{C})$, and we do so similarly with the parent cell $\mathcal{C}'$.

As the means of these random variables are equal to $P(\mathcal{C})$ and $P(\mathcal{C}')$ respectively, they are finite and sampled i.i.d, it follows from SLLN (theorem A.3) that

$$\left| \frac{|\mathcal{C} \cap S|}{|S|} - P(\mathcal{C}) \right| > \hat{p} - P(\mathcal{C})$$

happens finitely often, almost surely. Similarly, the following happens finitely often, almost surely.

$$\left| \frac{|\mathcal{C}' \cap S|}{|S|} - P(\mathcal{C}') \right| > P(\mathcal{C}') - \hat{p}$$

Finally the result follows from the fact that

$$\begin{aligned} &\left\{ \frac{|\mathcal{C} \cap S|}{|S|} > \hat{p} \right\} \cup \left\{ \frac{|\mathcal{C}' \cap S|}{|S|} < \hat{p} \right\} \\ \subseteq &\left\{ \left| \frac{|\mathcal{C} \cap S|}{|S|} - P(\mathcal{C}) \right| > \hat{p} - P(\mathcal{C}) \right\} \cup \left\{ \left| \frac{|\mathcal{C}' \cap S|}{|S|} - P(\mathcal{C}') \right| > P(\mathcal{C}') - \hat{p} \right\} \end{aligned}$$

□

**Remark 5.9.** *This actually leads to the observation that if we prove anything in limit using the true PHT, with only almost-sure guarantees, then we have the same conclusion using the sample PHT. This then ensures almost-sure asymptotic behavior of a limit using a PHT is identical under known and unknown distribution models.*

**Theorem 5.10.** *For two absolutely continuous probability measures $P_p, P_d$ over $[0,1]^d$, with p.d.fs $\phi_p, \phi_d$, we define the excess of a cell $\mathcal{C}_j$ to be*

$$\varepsilon_n(\mathcal{C}_j) = \left| \sum_{i=1}^{n} \mathbb{I}[X_i \in \mathcal{C}_j] - \sum_{i=1}^{n} \mathbb{I}[Y_i \in \mathcal{C}_j] \right| = \Big|\, |\mathcal{X} \cap \mathcal{C}_j| - |\mathcal{Y} \cap \mathcal{C}_j| \,\Big|$$

*where $\mathcal{X} = \{X_i\}$ and $\mathcal{Y} = \{Y_i\}$ are sampled i.i.d according to $P_p$ and $P_d$ respectively.*

*Then*

$$\lim_{n\to+\infty} \frac{1}{n} \sum_{j=1}^{N} \varepsilon_n(\mathcal{C}_j) \leq 2 \cdot d_{\mathrm{TV}}(P_p, P_d), \quad \textit{a.s.}$$

*Proof.* For some $j$,

$$\begin{aligned}
&\lim_{n\to+\infty} \left| \frac{\varepsilon_n(\mathcal{C}_j)}{n} - |P_p(\mathcal{C}_j) - P_d(\mathcal{C}_j)| \right| \\
&\leq \left| \lim_{n\to+\infty} \left( \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[X_i \in \mathcal{C}_j] - P_p(\mathcal{C}_j) \right) - \lim_{n\to+\infty} \left( \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[Y_i \in \mathcal{C}_j] - P_d(\mathcal{C}_j) \right) \right| \\
&= 0
\end{aligned}$$

almost surely by SLLN. Hence,

$$\begin{aligned}
\lim_{n\to+\infty} \frac{1}{n} \sum_{j=1}^{N} \varepsilon_n(\mathcal{C}_j) &= \sum_{j=1}^{N} |P_p(\mathcal{C}_j) - P_d(\mathcal{C}_j)| = \sum_{j=1}^{N} \left| \int_{\mathcal{C}_j} (\phi_p - \phi_d)\, \mathrm{d}\mathbf{r} \right| \\
&\leq \sum_{j=1}^{N} \int_{\mathcal{C}_j} |\phi_p - \phi_d|\, \mathrm{d}\mathbf{r} = \int_{[0,1]^d} |\phi_p - \phi_d|\, \mathrm{d}\mathbf{r} \\
&= 2 \cdot d_{\mathrm{TV}}(P_p, P_d)
\end{aligned}$$

□

We can say more about how we expect excess to behave in the special case of $P_p = P_d$, but this deviates from the main course of discussion and so has been left to the appendix (theorem D.2).

## 5.3 Algorithms

### 5.3.1 PLG Algorithm

We first present the PLG algorithm (algorithm 7) a very simple linear-time algorithm.

**input :** a set of demands $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $n > 1$.
PHT decomposition $\mathcal{P} = \{\mathcal{C}_i\}_{i=1}^{N}$.
**output:** a Stacker Crane tour through $\mathcal{S}$.

1 **initialize** empty list of links (directed edges).
2 Add link $(\hat{x}, \hat{y})$ chosen at random from $\mathcal{S}$.
3 Mark $\hat{x}$ as visited.
4 Add the $n$ pickup-to-delivery links $x_i \to y_i$, $i = 1, \ldots, n$.
5 $(x, y) \leftarrow (\hat{x}, \hat{y})$.
6 **while** *there exist non-visited deliveries* **do**
7 Let $\mathcal{C}$ be the cell such that $y \in \mathcal{C}$.
8 **if** *there exists an non-visited delivery in* $\mathcal{C}$ **then**
9 Let $x' \in \mathcal{C}$ be such a delivery.
10 **else**
11 Let $x'$ be any non-visited delivery anywhere.
12 Add link $y \to x'$.
13 Mark $x'$ as visited.
14 $y \leftarrow y'$, where $(x', y') \in \mathcal{S}$.
15 Add link $y \to \hat{x}$.

**Algorithm 7:** Pseudocode for the PLG algorithm

It is trivial to see the correctness of this algorithm: We mark every visited node, ensuring that we only visit a pickup-delivery pair once, hence ensuring a path. By line 15, we ensure that the path becomes a cycle. By lines 6 to 14, we visit every pair either intercellularly or intracellularly, hence the produced cycle is a Stacker-Crane tour.

**Theorem 5.11.** *Let $P_p, P_d$ be absolutely continuous with p.d.fs $\phi_p, \phi_d$ that satisfy*

$$\max\{\operatorname{ess\,sup} \phi_p, \operatorname{ess\,sup} \phi_d\} < C$$

*then for $\epsilon > 0$ the PLG algorithm satisfies*

$$\lim_{n \to +\infty} \frac{\text{cost}[PLG]}{\text{cost}[\text{opt}]} \leq 1 + \frac{\epsilon + \sqrt{d}(1 - \frac{1}{d}) d_{\text{TV}}(P_p, P_d)}{\mathbb{E}[d(X, Y)] + d_{\text{W}}(P_p, P_d)}$$

*almost surely when using a PHT constructed on $\mathcal{X} \cup \mathcal{Y}$ with*

$$\hat{p} = \frac{\epsilon^d}{2^{d(d-1)} C^{d-1} d^{d/2}}$$

*Proof.* We have three costs associated with PLG: the pickup-delivery cost, the intracellular cost, and the intercellular cost.

The intercellular cost may be bounded above by charging the diameter of the space to every delivery excess edge, i.e., every delivery-pickup pair created between cells. This excess is precisely half of the excess given in theorem 5.10 as we are only considering cells of excess delivery, which is equal to excess pickup.

The intracellular cost is given by the total length within a cell for any matched pairs. The number of such pairs is given by the minimum of $|\mathcal{X} \cap \mathcal{C}_j|$ and $|\mathcal{Y} \cap \mathcal{C}_j|$, as any disconnected pair would have been connected by the algorithm. Hence the total side length is bounded by the total length of the body diagonal of the each cell side length (counting multiplicities), hence

$$\sqrt{d} \sum_{j=1}^{N} \min\{|\mathcal{X} \cap \mathcal{C}_j|, |\mathcal{Y} \cap \mathcal{C}_j|\} \ell_j$$

Next, given that

$$\min\{|\mathcal{X} \cap \mathcal{C}_j|, |\mathcal{Y} \cap \mathcal{C}_j|\} \leq \frac{|\mathcal{X} \cap \mathcal{C}_j| + |\mathcal{Y} \cap \mathcal{C}_j|}{2}$$

If we construct the PHT on the distribution given by $P_{pd} = (P_p + P_d)/2$ then we note that by SLLN, for samples $\mathcal{X}'$ and $\mathcal{Y}'$ we have

$$\lim_{n \to +\infty} \frac{\sqrt{d}}{n} \sum_{j=1}^{N} \frac{|\mathcal{X}' \cap \mathcal{C}_j| + |\mathcal{Y}' \cap \mathcal{C}_j|}{2} \ell_j = \sqrt{d} \sum_{j=1}^{N} P_{pd}(\mathcal{C}_j) \ell_j \overset{5.6}{<} \epsilon$$

almost surely.

Combining these results with the lower bound on opt given in [39],

$$\begin{aligned}
\lim_{n \to +\infty} \frac{\text{cost[PLG]}}{\text{cost[opt]}} &\leq \lim_{n \to +\infty} \frac{\displaystyle\sum_{i=1}^{n} d(X_i, Y_i) + \sqrt{d} \sum_{j=1}^{N} \min\{|\mathcal{X} \cap \mathcal{C}_j|, |\mathcal{Y} \cap \mathcal{C}_j|\} \ell_j + \sqrt{d} \sum_{j=1}^{N} \varepsilon_n(\mathcal{C}_j)}{\displaystyle\sum_{i=1}^{n} d(X_i, Y_i) + \inf_{\sigma \in \Pi_n} \sum_{i=1}^{n} d(Y_i, X_{\sigma(i)})} \\
&\leq \lim_{n \to +\infty} \frac{\mathbb{E}[d(X,Y)] + \epsilon + \sqrt{d} \cdot d_{\text{TV}}(P_p, P_d)}{\mathbb{E}[d(X,Y)] + d_{\text{W}}(P_p, P_d)} \\
&\overset{A.11}{\leq} 1 + \frac{\epsilon + \sqrt{d}(1 - \frac{1}{d}) d_{\text{TV}}(P_p, P_d)}{\mathbb{E}[d(X,Y)] + d_{\text{W}}(P_p, P_d)}
\end{aligned}$$

almost surely. □

**Corollary 5.12.** *If $P_p = P_d$ and as in theorem 5.11, then PLG is a linear-time asymptotically almost-surely near-optimal algorithm. That is*

$$\lim_{n \to +\infty} \frac{\text{cost}[PLG]}{\text{cost[opt]}} \leq 1 + O(\epsilon), \quad a.s.$$

*Proof.* $d_{\mathrm{W}}(P_p, P_d) = d_{\mathrm{TV}}(P_p, P_d) = 0$, hence

$$\lim_{n\to+\infty} \frac{\mathrm{cost[PLG]}}{\mathrm{cost[opt]}} \leq 1 + \frac{\epsilon}{\mathbb{E}[d(X,Y)]}, \quad \text{a.s.}$$

□

### 5.3.2 ASPLICE

**input :** a set of demands $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $\mathcal{X} = \{x_i\}$ and $\mathcal{Y} = \{y_i\}$, $n > 1$.
PHT decomposition over the convolution of pickup and delivery distributions, $\mathcal{P} = \{\mathcal{C}_i\}_{i=1}^{N}$.
**output:** a Stacker Crane tour through $\mathcal{S}$.

**initialize** empty list of links (directed edges).
Add the $n$ pickup-to-delivery links $x_i \to y_i$, $i = 1, \ldots, n$
**foreach** $\mathcal{C} \in \mathcal{P}$ **do**
    **foreach** *delivery $y$ in $\mathcal{C}$, with corresponding pickup $x$* **do**
        **if** *there is non-visited pickup $x' \neq x$* **then**
            Add link $y \to x'$.
            Mark $x'$ visited.
$\mathcal{N} \leftarrow$ non-visited pickups and deliveries.
Compute representatives $m_j \in \mathcal{C}_j$, for $j = 1, \ldots N$.
Compute a solution $\{f_{ij}\}$ to the Euclidean Transportation Problem over supply $|\mathcal{C}_i \cap \mathcal{N} \cap \mathcal{Y}|$ centered at $m_i$ and demand $|\mathcal{C}_j \cap \mathcal{N} \cap \mathcal{X}|$ centered at $m_j$.
**for** $i \leftarrow 1, \ldots, N$ **do**
    **for** $j \leftarrow 1, \ldots, N$ **do**
        **if** $f_{ij} > 0$ **then**
            **for** $k \leftarrow 1, \ldots, f_{ij}$ **do**
                Pick $\hat{d} \in \mathcal{C}_i \cap \mathcal{N}$ and $\hat{p} \in \mathcal{C}_j \cap \mathcal{N}$.
                Add edge $\hat{d} \to \hat{p}$.
                Mark $\hat{d}$ and $\hat{p}$ as visited.
$\{S_j\}_{i=1}^{N_n} \leftarrow$ subtours.
`MergeSubtours`$(\{S_j\}_{i=1}^{N_n})$

**Algorithm 8:** Pseudocode for the ASPLICE algorithm

**Remark 5.13.** *A **representative**, as utilized in algorithm 8, is any "sensible" point to use as a hub for the transportation problem. A simple representative would be the center of the*

*cell, giving:*

$$d(\mathbf{r}, m_j) \leq \frac{\sqrt{d}\ell_j}{2}, \quad \textit{for } \mathbf{r} \in \mathcal{C}_j \tag{5.2}$$

*Other candidate representatives include the centroid of excess points and the median of the excess points, but for the sake of analysis we just require that it at least satisfies eq.* (5.2).

**Theorem 5.14.** *Let $P_p, P_d$ be absolutely continuous with p.d.fs $\phi_p, \phi_d$ that satisfy*

$$\max\{\operatorname{ess\,sup} \phi_p, \operatorname{ess\,sup} \phi_d\} < C$$

*then ASPLICE is asymptotically near optimal almost surely.*

*Proof.* Suppose we map every point in each cell $\mathcal{C}_j$ to its representative $m_j$ and then compute the EBMP on the new set of points, then the cost of this new matching is at most the cost of the original EBMP and the lengths of the mappings. This follows directly from the triangle inequality.

Any EBMP with some $Y$ and $X$ at the same point $m_j$ could join such pairs without penalty: suppose that $Y \to \hat{X}$ and $\hat{Y} \to X$ are part of some minimum matching, then we may form another minimum matching that is identical except $Y \to X$ and $\hat{Y} \to \hat{X}$. This does not increase the matching, because

$$d(\hat{Y}, \hat{X}) + d(X, Y) = d(\hat{Y}, \hat{X}) \leq d(\hat{Y}, X) + d(X, Y) + d(Y, \hat{X}) = d(\hat{Y}, X) + d(Y, \hat{X})$$

Hence we may connect delivery-pickups edges at each $m_j$, and solve EBMP on excess only. If we denote the excess pickups and deliveries $\mathcal{X}_E$ and $\mathcal{Y}_E$ respectively, then the EBMP is equivalent to the Transportation Problem which may be written as an Integer Linear Program:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} d(m_i, m_j) x_{ij}, \\ \text{subject to} \quad & \sum_{i=1}^{N} x_{ij} = |\mathcal{Y}_E \cap \mathcal{C}_j|, \quad \sum_{j=1}^{N} x_{ij} = |\mathcal{X}_E \cap \mathcal{C}_i|, \\ & x_{ij} \in \mathbb{Z}, \quad x_{ij} \geq 0 \end{aligned}$$

Next, consider if we just ran SPLICE on the set of all excess pickup and deliveries, centered at their cells' representatives, then the number of tours is almost surely $O(\log n)$ ( [39, Lemma 4.3] related to theorem D.1). Hence, as we pair off all points centered at each $m_j$ randomly, each minimum matching is equiprobable so the number of tours $N_n$ is $O(\log n)$ almost surely.

Therefore,

$$\begin{aligned}
&\lim_{n\to+\infty} \frac{\text{cost[ASPLICE]}}{\text{cost[opt]}} \\
&\leq \lim_{n\to+\infty} \frac{\sum_{i=1}^{n} d(X_i, Y_i) + \sum_{j=1}^{N} [|\mathcal{X} \cap \mathcal{C}_j| + |\mathcal{Y} \cap \mathcal{C}_j|] \frac{\sqrt{d}}{2} \ell_j + N_n + \text{cost[EBMP]}}{\sum_{i=1}^{n} d(X_i, Y_i) + \inf_{\sigma\in\Pi_n} \sum_{i=1}^{n} d(Y_i, X_{\sigma(i)})} \\
&\leq \lim_{n\to+\infty} \frac{\text{cost}[SPLICE]}{\sum_{i=1}^{n} d(X_i, Y_i) + \inf_{\sigma\in\Pi_n} \sum_{i=1}^{n} d(Y_i, X_{\sigma(i)})} + \frac{\frac{\sqrt{d}}{n} \sum_{j=1}^{N} \frac{|\mathcal{X} \cap \mathcal{C}_j| + |\mathcal{Y} \cap \mathcal{C}_j|}{2} \ell_j}{\frac{1}{n}\sum_{i=1}^{n} d(X_i, Y_i) + \frac{1}{n} \inf_{\sigma\in\Pi_n} \sum_{i=1}^{n} d(Y_i, X_{\sigma(i)})} \\
&\overset{(*)}{\leq} 1 + \frac{\epsilon}{\mathbb{E}[d(X,Y)] + d_{\text{W}}(P_p, P_d)}
\end{aligned}$$

almost surely, where $(*)$ follows from the asymptotic optimality of SPLICE and by construction of $P_{pd} = \frac{1}{2}[P_p + P_d]$ (the same argument as given in proof of theorem 5.11). □

It should be noted that ASPLICE has a slower running time than using SPLICE with a $(1+\epsilon)$-approximate EBMP solver, such as [2] with runtime in $O(n^{1+\epsilon})$ which provides an expected $(1+\epsilon)$-approximation, which was even suggested in [39]. However, we have proven in almost-sure terms, not expected terms, and our approach lends itself to a shorter implementation. Indeed, the authors of [39] state that for their own experiments they use an LP solver for the EBMP, which is the main bottleneck. In comparison, using an LP solver for the Transportation Problem on excess is considerably faster, and is in fact the approach taken in our experiments.

## 5.4 Experimental Results

We consider the relative performance of the different algorithms on different types of distribution, where we restrict ourselves to $d = 2$.

We consider the cost of the Stacker-Crane tours produced by 3 different algorithms under different parameters:

1. SPLICE ($\hat{p} = 0.1$) : the baseline algorithm
2. PLG ($\hat{p} = 0.01$)
3. PLG ($\hat{p} = 0.01$)

4. ASPLICE ($\hat{p} = 0.01$)
5. ASPLICE ($\hat{p} = 0.1$)

We ran each experiment, varying the number of pickup-delivery pairs from 10 to 200, repeating each 10 times.

We also present the average ratio of the algorithms to the baseline, the SPLICE algorithm, this estimates

$$\mathbb{E}\left[\frac{\text{cost}[\mathbb{A}]}{\text{cost}[\text{SPLICE}]]}\right]$$

All simulations were run single threaded on a laptop computer with a 2.8 GHz processor with 16GB of RAM. The EBMP in SPLICE was solved using the Hungarian algorithm, whereas the Transportation Problem in ASPLICE was solved using an LP solver.

Creation of the PQT has been omitted from the reported run times, though this was negligible in comparison to the run time of any of the algorithms.

### 5.4.1 Uniform Distributions

We begin with comparing algorithms on the basis of cost and run time on points sampled uniformly from the unit square $[0, 1]^2$.

As can be seen in fig. 5.2, sampling uniformly generates a PQT that is similar to the true PQT, which would have identically sized squares.

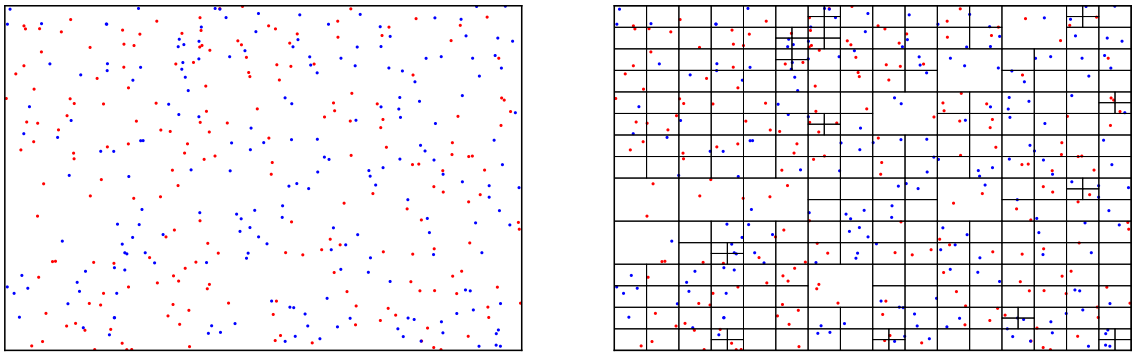


Figure 5.2: (a) 1000 sample pairs from a uniform distribution. (b) Point PQT generated on points with $\hat{p} < 0.1$.

The average cost for each number of pairs is reported in fig. 5.3(a) showing linear growth as we would expect given the lower bound analysis given in [39].

As we can see in fig. 5.3, these almost all quickly approach a constant value. The only algorithm's ratio that does not seem to be approaching a constant value is PLG with $\hat{p} = 0.01$,

though this is unsurprising as for small numbers of samples we would expect very few pickups or deliveries within a given cell, hence the randomly assigned intercellular delivery-pickup pairs would dominate.

We see that the average ratio for PLG and ASPLICE with $\hat{p} = 0.1$ initially have a large difference ($\sim 20\%$) but as the number of pairs increases this difference gets arbitrarily small. This is due to intracellular distance being gradually dominated by intercellular distance.

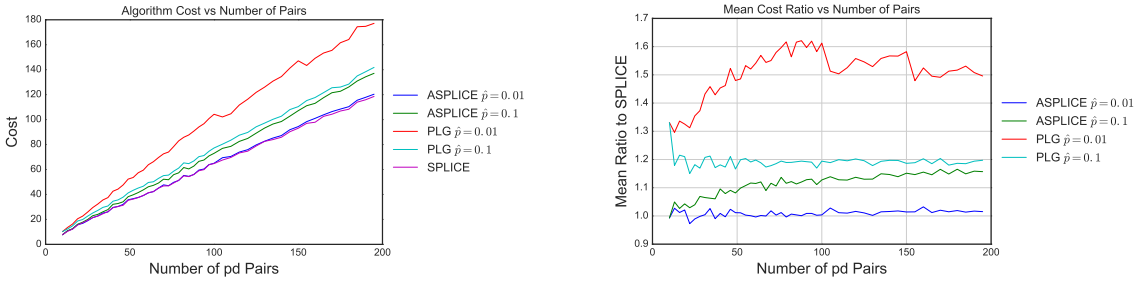



Figure 5.3: (a) Mean cost for different numbers of pairs. (b) Mean ratio of cost to SPLICE.

Finally we observe that ASPLICE ratio, with $\hat{p} = 0.01$, appears to stay around 1.01. The reason this is stays a good approximation of SPLICE is that $\hat{p} = 0.01$ will produce a very fine partition, so initially (at least the first 100 pairs) every point will be in its own cell, hence the transportation problem and Euclidean Bipartite Problem are equivalent. These runs then only differ by order of tour merge.

An important property of this algorithm may be observed in fig. 5.4 where, around 100 pairs, we see that the average run time of ASPLICE ($\hat{p} = 0.01$) stops growing with SPLICE. This indicates the algorithm is more likely to be placing points in non-empty cells hence we only change the supply/demand quantities, not the number of non-empty cells. This is beneficial as the Transportation Problem's run time grows slowly for increasing supply compared to adding vertices to EBMP, as occurs in SPLICE.

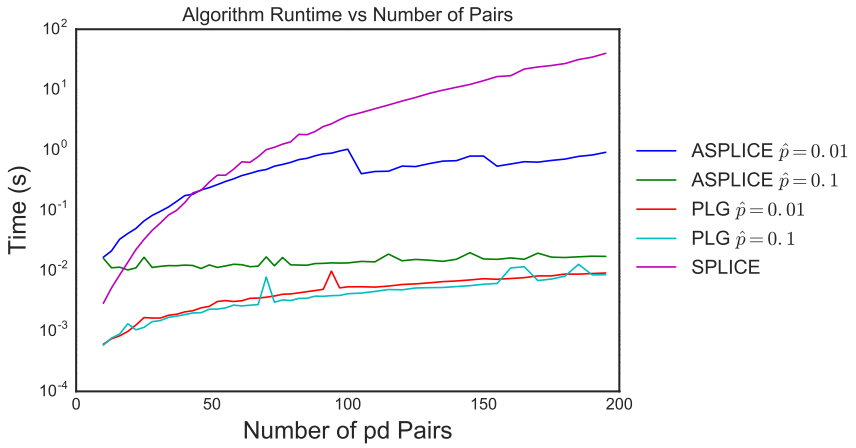



Figure 5.4: Plot of mean run time for the different algorithms. Note that the y-axis is logarithmically scaled.

### 5.4.2 Close Distributions

We next use a Gaussian Mixture Model with 3 components for the pickup distribution and use a slightly perturbed distribution for deliveries. In fig. 5.5 we can see a randomly generated sample PQT.

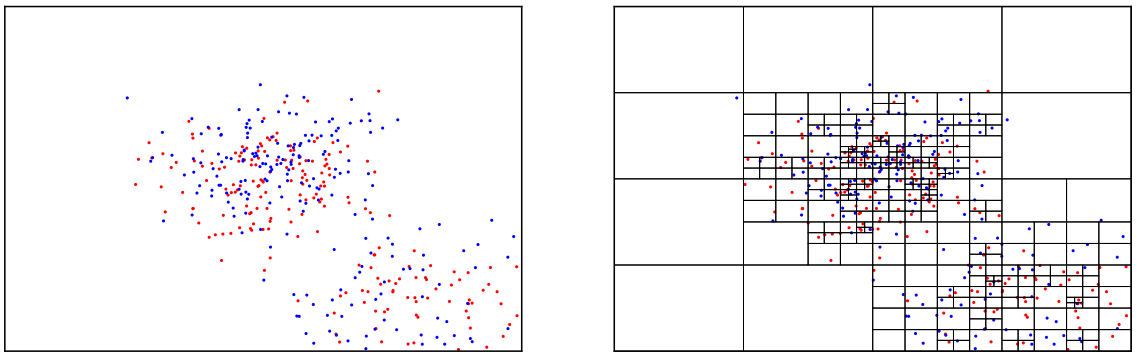


Figure 5.5: (a) Sample of points from two close pickup and delivery Gaussian Mixture Models. (b) Point PQT generated on points with $\hat{p} < 0.1$.

It is clear that we now have a difference in the asymptotic expected ratio cost for both PLG and ASPLICE ($\hat{p}$). This difference is due to the contribution of non-sublinear expected excess in cells (though with bound theorem 5.10).

We see similar run time results to the uniform distribution.

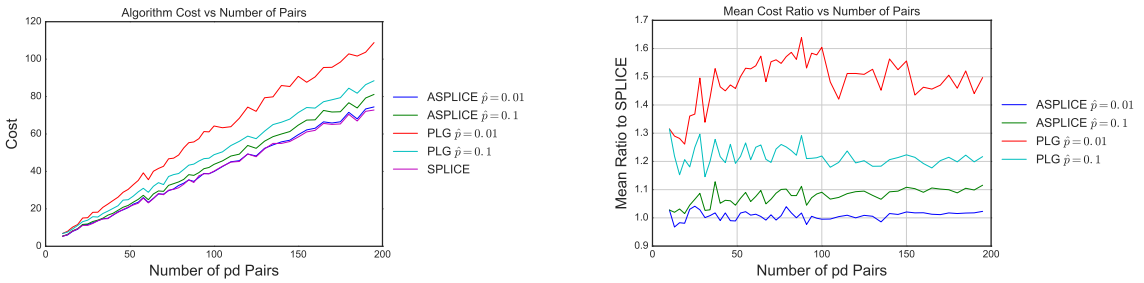



Figure 5.6: (a) Mean cost for different numbers of pairs. (b) Mean ratio of cost to SPLICE.

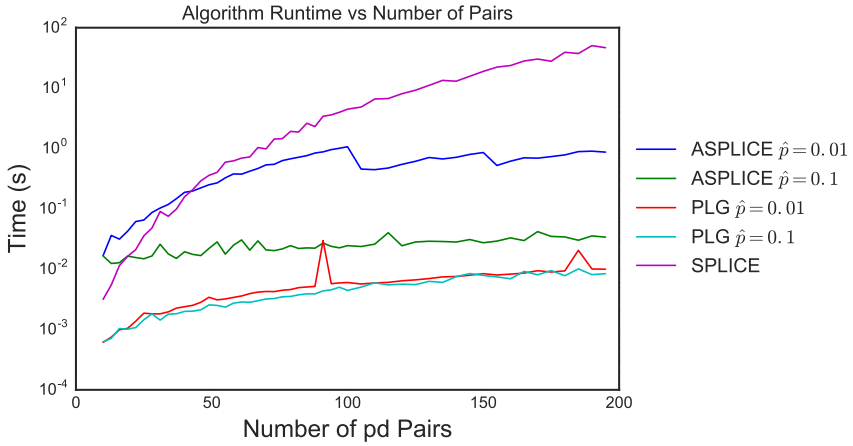



Figure 5.7: Plot of run time for the different algorithms. Note that the y-axis is logarithmically scaled.

### 5.4.3 Far Distributions

We next consider distributions that are far away in a Wasserstein-metric sense. This will increase the cost of the optimal solution (as proven in [39]).

Here we see that both PLG and ASPLICE ($\hat{p} = 0.1$) perform better than in the close case, however, the relative mean ratio gap is larger (+9.5%) compared to close (+9.0%).

The run time plot (fig. 5.10) is much the same as in the other experiments.

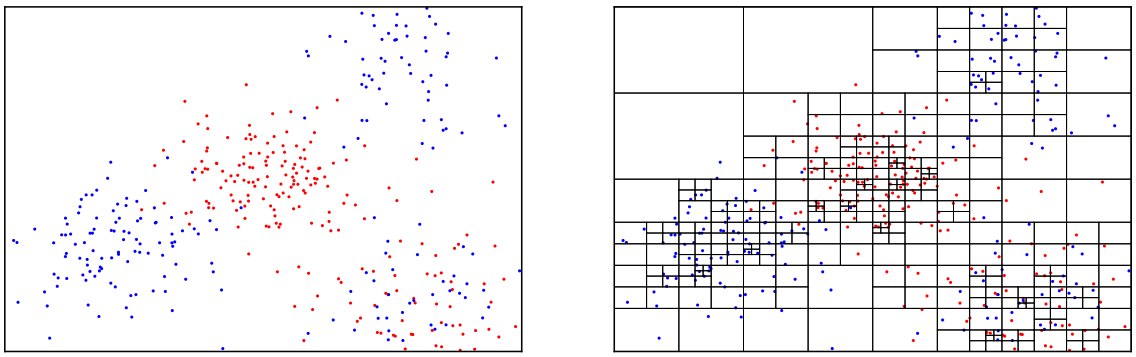


Figure 5.8: (a) Sample of points from two "far" pickup and delivery Gaussian Mixture Models. (b) Point PQT generated on points with $\hat{p} < 0.1$.

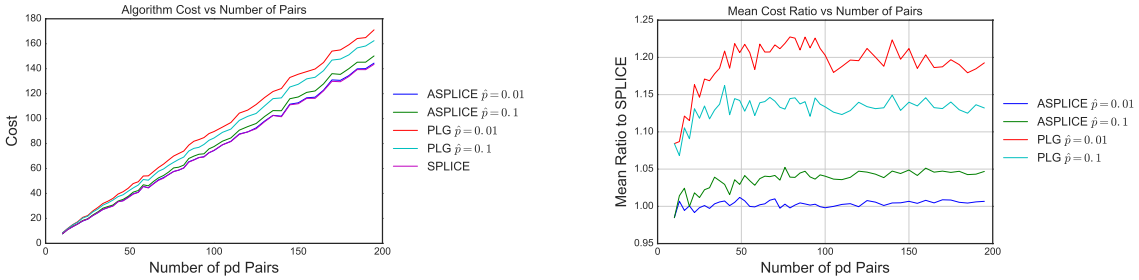



Figure 5.9: (a) Mean cost for different numbers of pairs. (b) Mean ratio of cost to SPLICE.

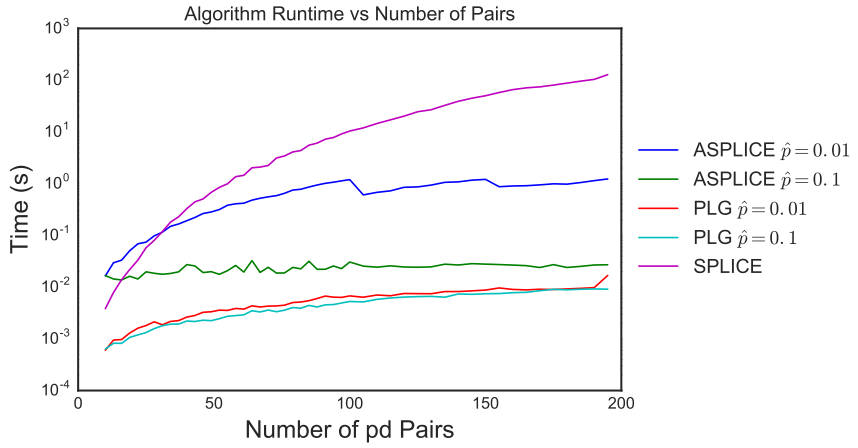



Figure 5.10: Plot of run time for the different algorithms. Note that the y-axis is logarithmically scaled.

# Chapter 6

# Conclusions

We considered different approach and algorithms to the stochastic $k$-server and Stacker-Crane problems. We first discussed the zoning and adaptive zoning algorithm, algorithms the we showed have bounded expected cost compared to an optimal algorithm under both known and unknown distribution models.

We then presented the lookahead algorithm for the stochastic $k$-server problem based on balancing both the greedy cost and the future expected minimum distance. We provided some methods to compute the expected minimum distance in special geometric cases using simplicial decompositions of Voronoi regions, before comparing the performance of lookahead to the greedy algorithm.

Finally we gave a linear-time asymptotically almost-surely near-optimal solution to the Stacker-Crane problem under the condition of equal absolutely-continuous distribution (with bounded p.d.fs) and also an approximation to SPLICE that offers asymptotically almost-sure near-optimality with a significant reduction in runtime.

These algorithms provide many different avenues of future research. Of particular importance, we hope to address the following:

1. Tighter lower bounds on the expected cost of an optimal algorithm on the stochastic $k$-server problem.

2. Extend the zoning algorithm analysis to weaker assumptions, such as to the Random Arrival Model, for which the optimal algorithm may know the requests ahead of time, but not the order in which they will be delivered.

3. Determine whether the condition of bounded p.d.fs may be weakened in the analysis of ASPLICE.

# Appendices

# Appendix A

# Miscellaneous Probability Theory

## A.1 Concentration Bounds

**Theorem A.1** (Multiplicative Chernoff Bound)**.** *Suppose $X_1, \ldots, X_n$ are independent random binary variables, $X$ denotes their sum, and $\mu = \mathbb{E}\left[X\right]$. Then*

$$\mathbb{P}[X \geq (1+\delta)\mu] \leq e^{-\delta^2\mu/3}, \quad 0 < \delta < 1, \tag{A.1}$$

$$\mathbb{P}[X \geq (1+\delta)\mu] \leq e^{-\delta\mu/3}, \quad 1 < \delta, \tag{A.2}$$

$$\mathbb{P}[X \leq (1-\delta)\mu] \leq e^{-\delta^2\mu/2}, \quad 0 < \delta < 1. \tag{A.3}$$

**Theorem A.2** (Hoeffding's Inequality [23])**.** *Suppose that $X_1, \ldots, X_n$ are independent random variables with $X_i \in [a_i, b_i]$ then*

$$\mathbb{P}\left(S_n - \mathbb{E}\left[S_n\right] \geq t\right) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right) \tag{A.4}$$

$$\mathbb{P}\left(\left|S_n - \mathbb{E}\left[S_n\right]\right| \geq t\right) \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right) \tag{A.5}$$

*where $S_n = X_1 + \cdots + X_n$.*

## A.2 Convergence Results

**Theorem A.3** (Strong Law of Large Numbers)**.** *Let $X_1, X_2, \ldots,$ be a sequence of i.i.d. random variables, each having finite mean $m$. Then*

$$\lim_{n\to\infty} \frac{X_1 + \cdots + X_n}{n} = m, \quad \text{a.s.}$$

**Lemma A.4** (Borel-Cantelli Lemma)**.** *Let* $A_1, A_2, \ldots \in \mathcal{F}$.

*i. If* $\sum_n \mathbb{P}(A_n) < \infty$, *then* $\mathbb{P}(\limsup_n A_n) = 0$.

*ii. If* $\sum_n \mathbb{P}(A_n) = \infty$, *and* $\{A_n\}$ *are indepedent, then* $\mathbb{P}(\limsup_n A_n) = 1$.

**Theorem A.5** ( [36, Lemma 5.2.1])**.** *Let* $X, X_1, X_2, \ldots$ *be random variables. Suppose for each* $\epsilon > 0$, *we have* $\mathbb{P}(|X_n - X| < \epsilon,\ i.o.) = 0$. *Then* $\mathbb{P}(X_n \to X) = 1$, *i.e.,* $\{X_n\}$ *converges to* $X$ *almost surely.*

**Theorem A.6** (de la Vallée-Poussin Theorem)**.** *A family* $\{X_\alpha\} \subset L^1(\mu)$ *is uniformly integrable if and only if there exists a non-negative increasing function* $G(t)$ *such that*

$$\lim_{t\to\infty} \frac{G(t)}{t} = \infty$$

*and*

$$\sup_\alpha \mathbb{E}\left[G(|X_\alpha|)\right] < \infty$$

**Theorem A.7.** *If a sequence of random variables* $\{X_i\}$ *that converges in distribution to* $X$ *is uniformly integrable, then* $\mathbb{E}\left[X_i\right] \to \mathbb{E}\left[X\right]$.

## A.3 Comparing Distributions

A very natural way to compare probability measures is to compare the maximum difference that may occur for any given event. This leads to a type of statistical distance that has numerous useful properties. The following is a simple extension of the definitions given in [28]:

**Definition A.8.** *For probability measures* $\mu, \nu$ *on* $\sigma$*-algebra* $\mathcal{F} \subset 2^\Omega$, *the* ***total variation distance of probability measures*** *is given by*

$$d_{\mathrm{TV}}(\mu, \nu) = \sup_{A\in\mathcal{F}} |\mu(A) - \nu(A)|$$

**Definition A.9.** *The* ***Wasserstein distance****,* $d_{\mathrm{W}}(\mu, \nu)$, *is given by*

$$d_{\mathrm{W}}(\mu, \nu) = \inf_{\gamma\in\Gamma(\mu,\nu)} \int_{M\times M} d(x, y)\, \mathrm{d}\gamma(x, y)$$

*where* $\Gamma$ *is the collection of all measures on* $M \times M$ *with marginals* $\mu$ *and* $\nu$.

**Theorem A.10.** *For a measurable space* $(\Omega, \mathcal{F})$, *if we have two probability measures* $\mu, \nu$ *that are absolutely continuous with respect to the Lebesgue measure* $\lambda$, *then*

$$d_{\mathrm{TV}}(\mu - \nu) = \frac{1}{2} \int_\Omega |\phi_\mu - \phi_\nu|\, \mathrm{d}\lambda$$

*where* $\phi_\mu$ *and* $\phi_\nu$ *denote the Radon-Nikodym derivatives.*

*Proof.* As probability measures are $\sigma$-finite, the Radon-Nikodym derivatives exist. Let $B = \{\, \mathbf{r} \mid \phi_\mu(\mathbf{r}) \geq \phi_\nu(\mathbf{r}) \,\}$. Given an arbitrary $A \in \mathcal{F}$,

$$\mu(A) - \nu(A) = \mu(A \cap B) - \nu(A \cap B) - \underbrace{[\nu(A \cap B^c) - \mu(A \cap B^c)]}_{\geq 0 \text{ by def. of } B^c} \leq \mu(A \cap B) - \nu(A \cap B)$$
$$\leq \mu(A \cap B) - \nu(A \cap B) + \underbrace{\mu(A^c \cap B) - \nu(A^c \cap B)}_{\geq 0} = \mu(B) - \nu(B)$$

we may similarly show that $\nu(A) - \mu(A) \leq \nu(B^c) - \mu(B^c)$. Given that

$$\nu(B^c) - \mu(B^c) = [1 - \nu(B)] - [1 - \mu(B)] = \mu(B) - \nu(B)$$

hence,

$$|\mu(A) - \nu(A)| \leq \mu(B) - \nu(B) = \nu(B^c) - \mu(B^c) = \tfrac{1}{2}[\mu(B) - \nu(B) + \nu(B^c) - \mu(B^c)]$$

where the equality occurs if $A = B$ or $A = B^c$.

$$\tfrac{1}{2}[\mu(B) - \nu(B) + \nu(B^c) - \mu(B^c)] = \tfrac{1}{2}\left[\int_B (\phi_\mu - \phi_\nu)\,\mathrm{d}\mathbf{r} + \int_{B^c} (\phi_\nu - \phi_\mu)\,\mathrm{d}\mathbf{r}\right]$$
$$= \tfrac{1}{2}\int_\Omega |\phi_\mu - \phi_\nu|\,\mathrm{d}\mathbf{r}$$

Hence,

$$d_{\mathrm{TV}}(\mu, \nu) = \tfrac{1}{2}\int_\Omega |\phi_\mu - \phi_\nu|\,\mathrm{d}\mathbf{r}$$

as we have only equalities, and $|\mu(A) - \nu(A)|$ is bounded with bound attained when $A = B$, then the supremum over $A \in \mathcal{F}$ is actually the maximum. □

There are numerous other types of probability distances that may be defined, e.g., [20].

**Theorem A.11.** *For probability measures $\mu, \nu$,*

$$d_{\mathrm{W}}(\mu, \nu) \leq \Delta \cdot d_{\mathrm{TV}}(\mu, \nu)$$

*where $\Delta$ is the diameter of the space.*

## A.4 Distributions

**Theorem A.12** (Normal Difference Distribution)**.** *Suppose we have two i.i.d random variables $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$, then their difference is distributed normally. Specifically,*

$$X - Y \sim \mathcal{N}(\mu_x - \mu_y, \sigma_x^2 + \sigma_y^2) \tag{A.6}$$

**Theorem A.13** (Folded Normal Distribution)**.** *Suppose that we have* $X \sim \mathcal{N}(0, \sigma_x^2)$*, then the folded distribution* $Y = |X|$ *satisfies*

$$\mu_y = \sigma_x \sqrt{\frac{2}{\pi}}, \qquad \sigma_y^2 = \sigma_x^2 \left(1 - \frac{2}{\pi}\right) \tag{A.7}$$

# Appendix B

# Algorithm Analysis

## B.1 Competitive Ratio

There are many ways to try and compare the behavior of different algorithms [22], but perhaps the most natural is to try and show that the cost of one algorithm always bounds another for a given initial configuration and sequence of requests. We typically compare an algorithm, online or not, to an offline algorithm Algorithm $\mathbb{A}$ run on a sequence of inputs $x$ and initial conditions $\mathcal{C}_0$, with a total cost given by cost $\left[\mathbb{A}(\mathcal{C}_0, x)\right]$ and the optimal algorithm $\mathbb{A}^*$ denoted $\text{opt}(\mathcal{C}_0, x) \equiv \mathbb{A}^*(\mathcal{C}_0, x)$.

**Definition B.1.** *An algorithm $\mathbb{A}$ is said to have a competitive ratio $\rho$ or to be $\rho$-competitive, if*

$$\text{cost}\left[\mathbb{A}(\mathcal{C}_0, x)\right] \leq \rho \cdot \text{cost}\left[\text{opt}(\mathcal{C}_0, x)\right] + \Phi(\mathcal{C}_0)$$

*for all $x$, where $\Phi$ is independent of $x$, and arguably more importantly it is independent of the length $|x|$.*

The issue in this definition is that if there is even one really bad sequence of requests then the competitive ratio could be very large. This could lead to a huge discrepancy between experimental use and the theoretical worst-case competitive ratio. For this reason we introduce the concept of **randomized algorithms**, algorithms where some part of the decision process was from a probability distribution, e.g., the choice between two new configurations may be decided by a coin toss.

We model our new type of analysis using the **oblivious adversary model**. We imagine a game in which we are competing our randomize online algorithm $\mathbb{A}$ with an adversary. The adversary wants to make algorithm $\mathbb{A}$ incur a large cost. The adversary knows everything about the algorithm including the probability distributions of any randomized decision. However, prior to generating the entire sequence of requests $x$, the outcome of any decision

is not known. The adversary then has to generate a sequence that maximizes the expected cost. Using this model we form the new competitive ratio

**Definition B.2.** *A randomized algorithm $\mathbb{A}$ is said to be $\rho$-competitive against an oblivious adversary if*

$$\mathop{\mathbb{E}}_{x}\left[\text{cost}\left[(\mathcal{C}_0, x)\right]\right] \leq \rho \cdot \text{cost}\left[\text{opt}(\mathcal{C}_0, x)\right] + \Phi(\mathcal{C}_0)$$

## B.2 Weakening Almost-Sure Convergence

While using expected cost provides some insight into the behavior of an algorithm, it does not tell us what occurs most of the time. There are a large number of related notions of asymptotic convergence and the nomenclature is still fairly non-standard. We shall use the definitions presented in [38]:

**Definition B.3.** *Given an event $E = E_n$ depending on a parameter $n$, we have five notions (in decreasing order of confidence) that an event is likely to hold:*

1. *An event $E$ holds surely if it is equal to the sure event $\bar{\emptyset}$*
2. *An event $E$ holds almost surely if it occurs with probability 1.*
3. *An event $E$ holds with overwhelming probability if, for every fixed $A > 0$, it holds with probability $1 - O(n^{-A})$, i.e., $\mathbb{P}(E) \geq 1 - C_A/n^A$ for some $C_A$ independent of $n$.*
4. *An event $E$ holds with high probability if it holds with probability $1 - O(1/n^c)$ for some $c > 0$ independent of $n$.*
5. *An event $E$ holds asymptotically almost surely if it holds with probability $1 - o(1)$, thus the probability of success goes to 1 in the limit $n \to \infty$.*

**Remark B.4.** *Asymptotically almost-surely is related to convergence in probability in the following way: If $|X_n - X| \leq \epsilon$ holds asymptotically almost surely for every $\epsilon > 0$.*

A summary of some basic results regarding high-probability type guarantees may be found in [25].

# Appendix C

# Miscellaneous Mathematical Results

## C.1 Orthants

If two variables $x$ and $y$ have the same sign then we greatly simplify the algebra and possibly final expression form when dealing with absolute values. This is because these variables may be added in a simple manner, i.e., $|x| + |y| = |x + y|$.

With the intent to extend this notion to the $L^1$ norm, we consider restrictions of a subset to certain regions of $\mathbb{R}^d$.

**Definition C.1** (Orthant)**.** *An orthant $\mathcal{O} \subset \mathbb{R}^d$ is given by*

$$\mathcal{O} = \{(x_1, \ldots, x_d) \in \mathbb{R}^d \mid \varepsilon_i x_i \geq 0,\ i = 1, \ldots, d\} \quad \textit{where } \varepsilon_i \in \{-1, 1\}$$

*The binary representation given by $b_1 b_2 \cdots b_d$, where $\varepsilon_i = (-1)^{b_i}$, uniquely identifies the orthant and hence there are $2^d$ such orthants.*

**Theorem C.2.** *A set $A \subset \mathbb{R}^d$ contained in an orthant satisfies*

$$\|x + y\|_1 = \|x\|_1 + \|y\|_1, \quad \textit{for } x, y \in A \tag{C.1}$$

*Proof.* As $x$ and $y$ are in a shared orthant there exists a shared correspoinding set $\{\varepsilon_i\} \in \{-1, 1\}^d$ as in C.1, then

$$\begin{aligned} \|x\|_1 + \|y\|_1 &= \sum_{i=1}^{d} |x_i| + \sum_{i=1}^{d} |y_i| = \sum_{i=1}^{d} \varepsilon_i x_i + \sum_{i=1}^{d} \varepsilon_i y_i \\ &= \sum_{i=1}^{d} \varepsilon_i (x_i + y_i) = \sum_{i=1}^{d} |x_i + y_i| = \|x + y\|_1 \end{aligned}$$

□

## C.2 Gamma Function

**Theorem C.3** (Bohr-Mollerup Theorem [5])**.** $\Gamma(x)$ *is the unique function that satisfies* $f(x+1) = xf(x)$*,* $f(1) = 1$ *and* $\log f(x)$ *is convex.*

**Remark C.4.** *The previous theorem gives a characterization of the gamma function that allows one to prove equivalency of the usual forms of the gamma function, e.g.,*

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}\,\mathrm{d}t = \lim_{n\to\infty}\frac{n!n^x}{x^{(n+1)}} = \frac{e^{-\gamma x}}{x}\prod_{n=1}^{\infty}\left(1+\frac{x}{n}\right)^{-1}e^{x/n}$$

*where* $x^{(n+1)}$ *denotes the rising factorial [ref] and* $\gamma$ *is the Euler-Mascheroni constant.*

By analytic continuation, $\Gamma$ may be defined for all but non-positive integers. This construction may be found in [19, Chapter 14].

**Theorem C.5.** *For any complex value* $t$*, the Gamma function* $\Gamma$ *satisfies*

$$\lim_{n\to+\infty}\frac{\Gamma(n+t)}{\Gamma(n)n^t} = 1 \tag{C.2}$$

**Theorem C.6** (Legendre Duplication Formula)**.** *For* $z \neq 0, -\frac{1}{2}, -\frac{2}{2}, -\frac{3}{2}, \ldots$ *we have*

$$\Gamma(z)\Gamma(z+\tfrac{1}{2}) = 2^{1-2z}\sqrt{\pi}\Gamma(2z) \tag{C.3}$$

# Appendix D

# Omitted Proofs

## D.1 PHT Section

The following is a known result from random permutation theory, with the asymptotic result given by [39, Lemma-4.3]. The first two results are standard calculations using a combinatorial generating function, however we present a proof that differs from that given in [37] which was referenced in [39].

**Theorem D.1.** *Suppose that $\mathcal{T}(\sigma)$ is the number of cycles in a permutation $\sigma \in \Pi_n$, if permutations are equiprobable, then*

$$\mathbb{E}\left[\mathcal{T}\right] = H_n \in \Theta(\log n) \tag{D.1}$$

*and*

$$\mathbb{V}\left[\mathcal{T}\right] = H_n - H_{n,2} \in \Theta(\log n) \tag{D.2}$$

*Furthermore,*

$$\lim_{n\to+\infty} \frac{\mathcal{T}(\sigma)}{n} = 0,\ \sigma \in \Pi_n, \quad a.s. \tag{D.3}$$

*Proof.* We will first consider the Stirling numbers of the First Kind $\begin{bmatrix} n \\ k \end{bmatrix}$, which serve as coefficients to the rising factorial $x^{(n)} = x(x+1)\cdots(x+n-1)$ [35],

$$x^{(n)} = \sum_{k=0}^{n} \begin{bmatrix} n \\ k \end{bmatrix} x^k$$

To establishes that the Sterling numbers of the first kind count the number of permutations containing exactly $k$ cycles in $\Pi_n$, consider what happens when we multiply the generating function of cycles of length $k$, by $x + n$. Multiplication by $x$ is analogous to just

appening a 1-cycle, increasing the cycle count by 1, e.g., $(1\,2)(3) \mapsto (1\,2)(3)(4)$. On the other hand multiplication by $n$ represents the unique permutations formed by inserting into any cycle after any element $i$, which does not change the cycle count, e.g., $(1\,2)(3) \mapsto \{(1\,4\,2)(3), (1\,2\,4)(3), (1\,2)(3\,4)\}$. Hence the equivalency follows from $x^{(n+1)} = x^{(n)}(x+n)$.

We may then calculate the moments for $\mathcal{T}(\sigma)$:

$$\mathbb{E}\left[\mathcal{T}^m\right] = \sum_{k=1}^{n} k^m \frac{\left[{n \atop k}\right]}{n!} = \frac{1}{n!} f_m(1)$$

where

$$f_m(x) = \sum_{k=1}^{n} k^m \left[{n \atop k}\right] x^k, \quad \text{which satisfies, } f_m(x) = x f'_{m-1}(x)$$

With the intention of finding a recursive form, we compute

$$f_1(x) = x f'_0(x) = x\frac{d}{dx}x^{(n)} = x\frac{d}{dx}\left[\frac{\Gamma(x+n)}{\Gamma(x)}\right] = x \cdot x^{(n)}[\psi^{(0)}(x+n) - \psi^{(0)}(x)] = f_0(x)g_1(x)$$

where $\psi^{(j)}(z)$ denotes the polygamma function [1].

Inductively, assume $f_{m-1}(x) = f_0(x)g_{m-1}(x)$,

$$\begin{aligned} f_m(x) &= x f'_{m-1}(x) = x\frac{d}{dx}[f_0(x)g_{m-1}(x)] \\ &= x f'_0(x)g_{m-1}(x) + x f_0(x)g'_{m-1}(x) \\ &= f_1(x)g_{m-1}(x) + x f_0(x)g'_{m-1}(x) \\ &= f_0(x)g_1(x)g_{m-1}(x) + x f_0(x)g'_{m-1}(x) = f_0(x)g_m(x) \end{aligned}$$

where $g_m(x) = g_1(x)g_{m-1}(x) + x g'_{m-1}(x)$.

Furthermore, given that $f_0(1) = 1^{(n)} = n!$,

$$\mathbb{E}\left[\mathcal{T}^m\right] = \frac{1}{n!} f_m(1) = g_m(1)$$

Consider the relation $\psi^{(j)}(z+1) = \psi^{(j)}(z) + (-1)^j \frac{j!}{z^{j+1}}$ (e.g., [1]), then we find

$$\psi^{(j)}(n+1) - \psi^{(j)}(1) = \sum_{i=1}^{n}[\psi^{(j)}(i+1) - \psi^{(j)}(i)] = (-1)^j j! \sum_{i=1}^{n} \frac{1}{i^{j+1}} = (-1)^j j! H_{n,j+1}$$

where $H_{n,a}$ is the partial sum of the series for $\zeta(a)$.

We may now calculate the first moment.

$$\mathbb{E}\left[\mathcal{T}\right] = g_1(1) = (1)[\psi^{(0)}(n+1) - \psi^{(0)}(1)] = H_{n,1} = H_n \in \Theta(\log n)$$

which is (D.1).

Similarly for the second moment:

$$\begin{aligned} g_2(x) = g_1(x)^2 + xg_1'(x) &= g_1(x)^2 + x\frac{d}{dx}\left(x[\psi^{(0)}(x+n) - \psi^{(0)}(x)]\right) \\ &= g_1(x)^2 + x^2[\psi^{(0)}(x+n) - \psi^{(0)}(x)] + x[\psi^{(1)}(x+n) - \psi^{(1)}(x)] \end{aligned}$$

We may now calculate,

$$\mathbb{E}\left[\mathcal{T}(\sigma)^2\right] = g_2(1) = H_n^2 + H_n + [\psi^{(1)}(n+1) - \psi^{(1)}(1)] = H_n^2 + H_n - H_{n,2}$$

$$\mathbb{V}\left[\mathcal{T}(\sigma)\right] = \mathbb{E}\left[\mathcal{T}(\sigma)^2\right] - \mathbb{E}\left[\mathcal{T}(\sigma)\right]^2 = H_n - H_{n,2} \in \Theta(\log n)$$

which follows as $H_{n,2} \to \zeta(2) = \pi^2/6$, whereas $H_n \in \Theta(\log n)$.

At this point we give more or less the same argument as [39, Lemma-4.3]. By the Chebychev inequality (theorem A.1),

$$\mathbb{P}\left[\frac{|\mathcal{T} - H_n|}{n} > \epsilon\right] \leq \frac{H_n - H_{n,2}}{\epsilon^2 n^2}$$

Using the fact that $\sqrt{n} > H_n$ for all $n > 2$,

$$\sum_{n=1}^{\infty} \mathbb{P}\left[\frac{|\mathcal{T} - H_n|}{n} > \epsilon\right] \leq \sum_{n=1}^{\infty} \frac{H_n - H_{n-2}}{\epsilon^2 n^2} < \frac{\zeta(3/2)}{\epsilon^2} < \infty$$

As this holds for all $\epsilon > 0$, theorem A.5

$$\lim_{n\to+\infty} \frac{|\mathcal{T}(\sigma) - H_n|}{n} = 0, \quad \text{a.s.}$$

Hence by the triangle inequality

$$0 \leq \lim_{n\to+\infty} \frac{|\mathcal{T}(\sigma)|}{n} \leq \lim_{n\to+\infty} \frac{|\mathcal{T}(\sigma) - H_n|}{n} + \lim_{n\to+\infty} \frac{|H_n|}{n} = 0, \quad \text{a.s.}$$

□

**Theorem D.2.** *Given two sequences of i.i.d Bernoulli random variables with probability p, $(X_i)_1^n$ and $(Y_i)_1^n$, we define the excess to be $\varepsilon_n = \sum_{i=1}^n X_i - \sum_{i=1}^n Y_i$. The expected absolute excess after n iterations has the following bound:*

$$\mathbb{E}\left[|\varepsilon_n|\right] \leq \frac{\Gamma\left(n + \frac{1}{2}\right)}{\sqrt{\pi}\Gamma(n)} \tag{D.4}$$

*where equality holds for* $p = \frac{1}{2}$.

*Furthermore,*

$$\lim_{n\to\infty} \frac{\mathbb{E}\left[|\varepsilon_n|\right]}{\sqrt{n}} = 2\sqrt{\frac{p(1-p)}{\pi}} \tag{D.5}$$

*Finally,*

$$\lim_{n\to+\infty} \frac{|\varepsilon_n|}{n} = 0, \quad a.s. \tag{D.6}$$

*Proof.* We perform a random walk on a lattice, incrementing $(1,0)$ with probability $p$ and incrementing $(0,1)$ similarly. A random walk ending on $(a,b)$ has the associated seminorm $\|(a,b)\| = |a-b|$ which, given that $a$ and $b$ represent the counts of increments, gives the absolute excess.

Define $D_{i-1} = |\varepsilon_i| - |\varepsilon_{i-1}|$, for $i = 1,\ldots,n$. Then by iterated expectation we obtain

$$\begin{aligned}\mathbb{E}\left[D_i\right] &= \mathbb{E}\left[\mathbb{E}\left[D_i \,\middle|\, |\varepsilon_i|\right]\right] = \sum_{j=0}^{n-1} \mathbb{E}\left[D_i \,\middle|\, |\varepsilon_i| = j\right] \mathbb{P}\left[|\varepsilon_i| = j\right] \\ &= \sum_{j=0}^{n-1} \mathbb{E}\left[D_i \,\middle|\, \|(a,b)\| = j\right] \mathbb{P}\left[\|(a,b)\| = j\right]\end{aligned} \tag{D.7}$$

Consider any random walk after $n-1$ iterations, that terminates on $(a,b)$. There are two possibilities:

1. If $a \neq b$.

   Due to symmetry we need only consider $a > b$. Given a further iteration, $D_i = 0$ with probability $p^2 + (1-p)^2$ when either both or none of the increments are made, $D_i = 1$ with probability $p(1-p)$ with increment in $(1,0)$ only, and $D_i = -1$ with probability $p(1-p)$ with increment in $(0,1)$ only. Hence

   $$\begin{aligned}\mathbb{E}\left[D_i \,\middle|\, \|(a,b)\| > 0\right] &= (0)(p^2 + (1-p)^2) + (1)(p(1-p)) + (-1)(p(1-p)) \\ &= 0\end{aligned}$$

2. If $a = b$.

   This represents the case of $(a,b)$ falling on the diagonal of the lattice. The seminorm is 0 so we get $D_i = 1$ as $\|(a+1,a)\| = \|(a,a+1)\| = 1 + \|(a,b)\|$ when only one increment occurs, which occurs with probabilty $2p(1-p)$. Hence,

   $$\begin{aligned}\mathbb{E}\left[D_i \,\middle|\, \|(a,b)\| = 0\right] &= (0)(p^2 + (1-p)^2) + (1)2p(1-p) \\ &= 2p(1-p)\end{aligned} \tag{D.8}$$

By *eq.* (D.7) we obtain

$$\mathbb{E}\left[D_i\right] = 2p(1-p)\,\mathbb{P}\left[\|(a,b)\| = 0\right] = 2p(1-p)\sum_{j=0}^{i}\mathbb{P}\left[a=b=j\right]$$

The number of random walks with $(j,j)$ after the $i$ iterations is given by the number of ways that a walk can have $j$ steps out of $i$ possible in both the $(1,0)$ and $(0,1)$ directions, i.e., $\binom{i}{j}^2$ possible walks. The probability of any such random walks is $p^{2j}(1-p)^{2i-2j}$.

Clearly $\mathbb{E}[|\varepsilon_0|] = 0$, so by linearity of expectation

$$\mathbb{E}[|\varepsilon_n|] = \sum_{i=0}^{n-1}\mathbb{E}[D_i] = 2p(1-p)\sum_{i=0}^{n-1}\sum_{j=0}^{i}\binom{i}{j}^2 p^{2j}(1-p)^{2i-2j} \tag{D.9}$$

This may be seen to be maximized at $p = 1/2$.

$$2(\tfrac{1}{2})(1-\tfrac{1}{2})\sum_{i=0}^{n-1}\sum_{j=0}^{i}\binom{i}{j}^2(\tfrac{1}{2})^{2j}(1-\tfrac{1}{2})^{2i-2j} = \tfrac{1}{2}\sum_{i=0}^{n-1}\frac{1}{4^i}\sum_{j=0}^{i}\binom{i}{j}^2 = \tfrac{1}{2}\sum_{i=0}^{n-1}\frac{\binom{2i}{i}}{4^i} = \frac{n\binom{2n}{n}}{4^n} \tag{D.10}$$

The latter may be seen by induction, with the inductive step using the fact that the central binomial coefficient satisfies $\binom{2n}{n} = \frac{n+1}{2(2n+1)}\binom{2(n+1)}{n+1}$:

$$\frac{n\binom{2n}{n}}{4^n} + \frac{1}{2}\frac{\binom{2n}{n}}{4^n} = \frac{2n+1}{2\cdot 4^n}\binom{2n}{n} = \frac{2n+1}{2\cdot 4^n}\frac{n+1}{2(2n+1)}\binom{2(n+1)}{n+1} = \frac{(n+1)\binom{2(n+1)}{n+1}}{4^{n+1}}$$

By eq. (C.3) we have

$$\binom{2n}{n} = \frac{\Gamma(2n+1)}{\Gamma(n+1)^2} = \frac{(2n)\Gamma(2n)}{n^2\Gamma(n)^2} = 2\frac{\Gamma(n)\Gamma(n+\frac{1}{2})}{2^{1-2n}\sqrt{\pi}n\Gamma(n)^2} = 4^n\frac{\Gamma(n+\frac{1}{2})}{\sqrt{\pi}n\Gamma(n)} \tag{D.11}$$

Combining eqs. (D.10) and (D.11) gives eq. (D.4).

Consider on the other hand, the central limit theorem gives that

$$\frac{\sum_{i=1}^{n}X_i - \mu n}{\sqrt{np(1-p)}} \xrightarrow{d} \mathcal{N}(0,1), \qquad \frac{\sum_{i=1}^{n}Y_i - \mu n}{\sqrt{np(1-p)}} \xrightarrow{d} \mathcal{N}(0,1)$$

then

$$\frac{|\varepsilon_n|}{\sqrt{np(1-p)}} = \left|\frac{\sum_{i=1}^{n}X_i - \sum_{i=1}^{n}Y_i}{\sqrt{np(1-p)}}\right| = \left|\frac{\sum_{i=1}^{n}X_i - \mu n}{\sqrt{np(1-p)}} - \frac{\sum_{i=1}^{n}Y_i - \mu n}{\sqrt{np(1-p)}}\right| \tag{D.12}$$

which converges weakly to a Half-Normal distribution, with mean $\frac{2}{\sqrt{\pi}}$ and variance $2\left(1-\frac{2}{\pi}\right)$. This may be derived from the fact that eq. (D.12) has a weak limit of a difference of normal random variables eq. (A.6) that is folded into a Half-Normal distribution eq. (A.7).

We intend to strenthen convergence in distribution to convergence of means. By theorem A.7, this may be done by proving uniform integrability of the sequence $\left\{\frac{|\varepsilon_n|}{\sqrt{np(1-p)}}\right\}$.

Let $Z_i = X_i - Y_i$, then clearly $\varepsilon_i = \varepsilon_{i-1} + Z_{i-1}$, so

$$\mathbb{E}\left[|\varepsilon_i|^2\right] = \mathbb{E}\left[\varepsilon_i^2\right] = \mathbb{E}\left[\varepsilon_{i-1}^2\right] + 2\,\mathbb{E}\left[\varepsilon_{i-1}\right]\mathbb{E}\left[Z_{i-1}\right] + \mathbb{E}\left[Z_{i-1}^2\right]$$

Given that $X_{i-1}$ and $Y_{i-1}$ are i.i.d, $\mathbb{E}\left[Z_{i-1}\right] = 0$. Furthermore, $(Z_{i-1})^2 = 1$ if $x_{i-1} = 1, y_{i-1} = 0$ or $x_{i-1} = 0, y_{i-1} = 1$, each of which occurs with probability $p(1-p)$ and 0 otherwise. Therefore $\mathbb{E}\left[Z_{i-1}^2\right] = 2p(1-p)$, and

$$\mathbb{E}\left[|\varepsilon_n|^2\right] = \mathbb{E}\left[|\varepsilon_{n-1}|^2\right] + 2p(1-p) = \mathbb{E}\left[|\varepsilon_0|^2\right] + 2np(1-p) = 2np(1-p)$$

It follows that

$$\sup_n \mathbb{E}\left[\left|\frac{\varepsilon_n}{\sqrt{np(1-p)}}\right|^2\right] = \sup_n \frac{1}{np(1-p)}\,\mathbb{E}\left[|\varepsilon_n|^2\right] = 2 < \infty$$

Applying theorem A.6 with $G(t) = t^2$ gives uniform integrability and therefore eq. (D.5) follows.

To finish we need to show (D.6). We notice that $\varepsilon_n = \sum_{i=1}^n Z_i$, $Z_i \in [-1, 1]$, and $\mathbb{E}\left[\varepsilon_i\right] = i\,\mathbb{E}\left[Z_i\right] = 0$, then by Hoeffding's inequality theorem A.2

$$\mathbb{P}\left[\frac{|\varepsilon_n|}{n} > \epsilon\right] = \mathbb{P}\left[|\varepsilon_n - \mathbb{E}\left[\varepsilon_n\right]| > n\epsilon\right] < 2\exp\left(-\frac{n\epsilon^2}{2}\right)$$

for any $\epsilon > 0$.

Then,

$$\sum_{n=1}^{\infty} \mathbb{P}\left[\frac{|\varepsilon_n|}{n} > \epsilon\right] < \sum_{n=1}^{\infty} 2\exp\left(-\frac{n\epsilon^2}{2}\right) = \frac{2}{e^{\epsilon^2/2} - 1} < \infty$$

where the series is evaluated as a geometric series. By the Borel-Cantelli lemma A.4 it follows that $\mathbb{P}\left[\limsup_{n\to\infty} \frac{|\varepsilon_n|}{n} > \epsilon\right] = 0$. As this holds for any $\epsilon > 0$ then by theorem A.5 we conclude $|\varepsilon_n|/n \to 0$ almost surely. □

# Bibliography


[1] Milton Abramowitz, Irene A Stegun, et al. Handbook of mathematical functions. *Applied mathematics series*, 55:62, 1966.

[2] Pankaj Agarwal and Kasturi Varadarajan. A near-linear constant-factor approximation for euclidean bipartite matching? In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 247–252. ACM, 2004.

[3] Pankaj K Agarwal, Alon Efrat, and Micha Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM Journal on Computing*, 29(3):912–953, 2000.

[4] Aris Anagnostopoulos, Clément Dombry, Nadine Guillotin-Plantard, Ioannis Kontoyiannis, and Eli Upfal. Probabilistic analysis of the k-server problem on the circle. In *21st International Meeting on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (AofA 2010)*, 2010.

[5] Emil Artin. *The gamma function*. Courier Dover Publications, 2015.

[6] David S Atkinson and Pravin M. Vaidya. Using geometry to solve the transportation problem in the plane. *Algorithmica*, 13(5):442–461, 1995.

[7] Nikhil Bansal, Niv Buchbinder, Aleksander Madry, et al. A polylogarithmic-competitive algorithm for the k-server problem. *arXiv preprint arXiv:1110.1580*, 2011.

[8] Yair Bartal and Eddie Grove. The harmonic k-server algorithm is competitive. *Journal of the ACM (JACM)*, 47(1):1–15, 2000.

[9] Cristina Bazgan, Refael Hassin, and Jérôme Monnot. Approximation algorithms for some vehicle routing problems. *Discrete Applied Mathematics*, 146(1):27–42, 2005.

[10] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition, 2008.

[11] Christian Buchta. Zufallspolygone in konvexen vielecken. *Journal für die reine und angewandte Mathematik*, 347:212–220, 1984.

[12] L Paul Chew and Robert L Scot Dyrsdale III. Voronoi diagrams based on convex distance functions. In *Proceedings of the first annual symposium on Computational geometry*, pages 235–244. ACM, 1985.

[13] Artur Czumaj and Christian Sohler. Sublinear-time approximation for clustering via random sampling. *Automata, Languages and Programming*, pages 41–52, 2004.

[14] René descartes. *Principia philosophiae*. apud Ludovicum Elzevirium, 1994.

[15] Amos Fiat, Richard M Karp, Michael Luby, Lyle A McGeoch, Daniel D Sleator, and Neal E Young. Competitive paging algorithms. *Journal of Algorithms*, 12(4):685–699, 1991.

[16] Amos Fiat, Yuval Rabani, and Yiftach Ravid. Competitive k-server algorithms. In *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*, pages 454–463. IEEE, 1990.

[17] Steven Fortune. A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2(1-4):153–174, 1987.

[18] Greg N Frederickson, Matthew S Hecht, and Chul E Kim. Approximation algorithms for some routing problems. In *Foundations of Computer Science, 1976., 17th Annual Symposium on*, pages 216–227. IEEE, 1976.

[19] Theodore Gamelin. *Complex analysis*. Springer Science & Business Media, 2003.

[20] Alison L Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International statistical review*, 70(3):419–435, 2002.

[21] Julian Havil and J Gamma. Exploring eulers constant, 2003.

[22] Benjamin Hiller and Tjark Vredeveld. Probabilistic alternatives for competitive analysis. *Computer Science-Research and development*, 27(3):189–196, 2012.

[23] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[24] Abdolhossein Hoorfar and Mehdi Hassani. Inequalities on the lambert w function and hyperpower function. *J. Inequal. Pure and Appl. Math*, 9(2):5–9, 2008.

[25] dennis Komm, Rastislav Královič, Richard Královič, and Tobias Mömke. Randomized online computation with high probability guarantees. *arXiv preprint arXiv:1302.2805*, 2013.

[26] Elias Koutsoupias. The k-server problem. *Computer Science Review*, 3(2):105–118, 2009.

[27] Elias Koutsoupias and Christos H Papadimitriou. Beyond competitive analysis. *SIAM Journal on Computing*, 30(1):300–317, 2000.

[28] David Asher Levin, Yuval Peres, and Elizabeth Lee Wilmer. *Markov chains and mixing times*. American Mathematical Soc., 2009.

[29] Bruno Lévy and Yang Liu. Lp centroidal voronoi tessellation and its applications. In *ACM Transactions on Graphics (TOG)*, volume 29, page 119. ACM, 2010.

[30] Mark S Manasse, Lyle A McGeoch, and Daniel D Sleator. Competitive algorithms for server problems. *Journal of Algorithms*, 11(2):208–230, 1990.

[31] Lyle A McGeoch and Daniel D Sleator. A strongly competitive randomized paging algorithm. *Algorithmica*, 6(1-6):816–825, 1991.

[32] Kari S McLeod. Our sense of snow: the myth of john snow in medical geography. *Social Science & Medicine*, 50(7):923–935, 2000.

[33] Adam Meyerson, Liadan O'callaghan, and Serge Plotkin. A k-median algorithm with running time independent of data size. *Machine Learning*, 56(1-3):61–87, 2004.

[34] Christos H Papadimitriou. The euclidean travelling salesman problem is np-complete. *Theoretical Computer Science*, 4(3):237–244, 1977.

[35] Steven Roman. *The umbral calculus*. Springer, 2005.

[36] Jeffrey Seth Rosenthal. *A first look at rigorous probability theory*. World Scientific, 2006.

[37] LA Shepp and S_P Lloyd. Ordered cycle lengths in a random permutation. *Transactions of the American Mathematical Society*, 121(2):340–357, 1966.

[38] Terence Tao. 254a, notes 0: A review of probability theory, Jan 2010.

[39] Kyle Treleaven, Marco Pavone, and Emilio Frazzoli. Asymptotically optimal algorithms for one-to-one pickup and delivery problems with applications to transportation systems. *IEEE Transactions on Automatic Control*, 58(9):2261–2276, 2013.

[40] Jianan Wang. Constructing voronoi diagrams in l(1) and l(infinity) metrics with two plane sweeps. Master's thesis, University of Windsor, Ottawa, Ontario, Canada, 1994.

[41] Lei Zhang and Rui Zhang. Polynomial heuristics for the k-stacker crane problem. In *Systems, Man and Cybernetics, 1993.'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on*, pages 266–268. IEEE, 1993.