A Computational Exploration of Rounding Algorithms for the Multiway Cut Problem

by

Damon Mosk-Aoyama

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Bachelor of Science in Computer Science and Engineering

and

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2002

Author . . . .
Department of Electrical Engineering and Computer Science
May 28, 2002

Certified by . . . .
David R. Karger
Associate Professor
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

A Computational Exploration of Rounding Algorithms for the Multiway Cut Problem

by

Damon Mosk-Aoyama

Submitted to the Department of Electrical Engineering and Computer Science
on May 28, 2002, in partial fulfillment of the
requirements for the degrees of
Bachelor of Science in Computer Science and Engineering
and
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The goal of the multiway cut problem is to find a minimum-cost set of edges whose removal disconnects a certain set of k distinguished vertices in a graph. Călinescu, Karloff, and Rabani gave a geometric relaxation of the multiway cut problem and a rounding scheme that yield an approximation algorithm for the problem that has an approximation ratio of $3/2 - 1/k$. In subsequent work, Karger, Klein, Stein, Thorup and Young improved the approximation ratio of the algorithm by using different rounding schemes, which they found through the assistance of computational experiments. Their rounding schemes are based on a single class of partitions of the k-simplex. We extend their computational experiments by introducing additional types of partitions of the 4-simplex. Our experimental results suggest that an optimal rounding scheme for $k > 3$ cannot be defined purely in terms of the partitions of the simplex that Karger et al. studied. Furthermore, we specify a rounding scheme for $k = 4$ that improves upon the best approximation ratio for which an analytic proof has been given.

Thesis Supervisor: David R. Karger
Title: Associate Professor

Acknowledgments

I would like to thank my parents Sachiyo and Carl and my brother Simon. Their support over the past years has been vital to me.

During the course of this project, I benefitted from several discussions with Brian Dean. I would like to thank Brian for the ideas and the effort that he contributed to this work.

Finally, I must thank David Karger. In addition to advising this thesis, he has given me numerous opportunities to work on other interesting problems. I am grateful for his enthusiasm and advice.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the multiway cut problem, the goal is to find a minimum-cost set of edges whose removal disconnects a certain set of k distinguished vertices in a graph. The technique of using a geometric embedding to develop an approximation algorithm for a graph optimization problem was successfully employed for the multiway cut problem by Călinescu, Karloff, and Rabani, who gave a geometric relaxation of the problem and a rounding scheme that lead to an algorithm with an approximation ratio of $3/2 - 1/k$. Subsequently, Karger, Klein, Stein, Thorup and Young showed that the approximation ratio of the algorithm may be improved through the use of different rounding schemes, which they found through the assistance of computational experiments. All of the rounding schemes they consider are based on one class of partitions of the k-simplex.

In the present work, we extend the computational experiments of Karger et al. by introducing several additional types of partitions of the 4-simplex. The results of our experiments suggest that an optimal rounding scheme for $k > 3$ cannot be defined purely in terms of the partitions of the simplex that Karger et al. studied. We also specify a rounding scheme for $k = 4$ that yields an approximation ratio better than the smallest known ratio for which an analytic proof has been given.

1.1 Problem Definition

The minimum (s, t)-cut problem is a classic graph-optimization problem. Given an undirected graph $G = (V, E)$ in which the edges have nonnegative costs, and two distinct vertices $s, t \in V$, an (s, t)-cut is a subset of the edges satisfying the property that if the edges in the cut are removed from the graph, s and t are in different connected components of the resulting graph. Such a cut is said to *separate* or *disconnect* the two vertices s and t. The cost of a particular (s, t)-cut is defined as the sum of the costs of the edges in the cut. In the minimum (s, t)-cut problem, the inputs are a graph and two vertices s and t in the graph, and the goal is to find an (s, t)-cut of the graph with the minimum cost.

This problem has a natural generalization known as the multiway cut problem that involves more than two distinguished vertices. The inputs in the multiway cut problem are a graph $G = (V, E)$ and a subset $T \subseteq V$ of the vertices. Each vertex in T is referred to as a terminal. Given that there are $|T| = k$ terminals, a k-way cut of the graph is a set of edges that, when removed, will produce a graph in which each of the k terminals is in a distinct connected component. As in the minimum (s, t)-cut problem, the cost of a cut is the sum of the costs of the edges in the cut, and the goal of the multiway cut problem is to find a k-way cut with minimum cost in the graph. The minimum (s, t)-cut problem is the special case $k = 2$ of the more general multiway cut problem.

The multiway cut problem arises in several applications. One may formulate the scheduling of jobs on multiple processors in distributed computing systems as a multiway cut problem [5]. Other applications include partitioning files among the nodes of a network, assigning users to computers in an environment with multiple computers, and partitioning the elements of a circuit during the design of electronic chips into the subcircuits that are placed on different chips [2]. Note that the standard multiway cut problem does not require the connected components that remain in the graph after the edges in the cut have been removed to be balanced in the sense that their relative sizes under some measure must satisfy additional constraints. Placing such a requirement on the connected components would produce a more general balanced multiway cut problem that would have more potential applications than the standard multiway cut problem.

1.2 Prior Work

The multiway cut problem was first studied by Dahlaus, Johnson, Papadimitriou, Seymour, and Yannakakis [2]. They showed that for all fixed $k \geq 3$, multiway cut is MAX-SNP hard, and that multiway cut is NP-hard even when the costs of all of the edges are fixed at 1. Studies of the multiway cut problem have therefore focused on the development of approximation algorithms for the problem. In the context of multiway cut, an approximation algorithm for the problem is said to have an approximation ratio of α if, for any input, the ratio of the cost of the k-way cut produced by the algorithm to the cost of the minimum k-way cut for the input is at most α. We seek an algorithm with the smallest possible approximation ratio, though of course there is no algorithm that has an approximation ratio less than 1. The fact that multiway cut is MAX-SNP hard implies that there exists some constant $b > 1$ such that no polynomial-time approximation algorithm for multiway cut can have an approximation ratio less than b unless P = NP.

In addition to demonstrating the hardness of approximation of multiway cut, Dahlaus et al. gave the first approximation algorithm for the problem. This algorithm is based on an isolation heuristic. For each terminal in the graph, it uses an algorithm for the minimum (s,t)-cut problem to find the minimum-cost cut separating the terminal from the remaining $(k-1)$ terminals. The algorithm outputs the set of edges obtained by forming the union of the $(k-1)$ smallest-cost cuts. Because each of the $(k-1)$ cuts, when removed from the graph, eliminates all paths between one of the terminals and any other terminal, removing the union of the cuts from the graph will leave no paths between any pair of terminals in the graph. Dahlaus et al. show that this algorithm achieves an approximation ratio of $2 - 2/k$.

The Dahlaus et al. approximation algorithm was the best known approximation algorithm for the general multiway cut problem on arbitrary graphs until the work of Călinescu, Karloff, and Rabani [1]. They developed a new approach to finding a k-way cut that approximates the minimum. At the core of their algorithm is a geometric embedding of the input graph into the k-simplex. The nodes of the graph are mapped to points in the simplex, with the terminals mapping to the k vertices of the simplex.

Once the graph has been embedded into the simplex, the simplex is partitioned into regions, one for each terminal. A partition of the simplex naturally induces a k-way cut of the graph, with the nodes in the connected component that contains each terminal mapping to points in the region for that terminal. In this context, an edge in the input graph is in the cut if its endpoints map to different regions of the partitioned simplex. Călinescu et al. exploit this geometric embedding of the graph into the simplex to give an approximation algorithm for multiway cut that has an approximation ratio of $3/2 - 1/k$.

Furthering the study of the Călinescu et al. approximation algorithm, Karger, Klein, Stein, Thorup, and Young showed that by modifying the way in which the algorithm partitions the simplex, its approximation ratio can be improved [4]. Their work is based on computational experiments involving probability distributions over a class of partitions of the simplex that they refer to as *side-parallel cuts*, abbreviated *sparcs*. For the case of $k = 3$ terminals, they present a probability distribution over partitions of the simplex that, when used in the Călinescu et al. algorithm, leads to an approximation ratio that is the best possible for any algorithm based on the Călinescu et al. framework. In this sense, their work demonstrates the limits of the Călinescu et al. method for the case of $k = 3$ terminals. Karger et al. also give a probability distribution over sparcs that leads to improved approximation ratios for all $k > 3$, but do not show that this distribution is optimal.

1.3 Our Approach

In this project, we use computational experiments to study approximation algorithms for the multiway cut problem. Our approach is to extend the work of Karger et al. by considering alternatives to sparcs for partitioning the simplex. The types of cuts that we introduce in addition to sparcs are defined based on the 4-simplex, a regular tetrahedron, and as such our work focuses on the case $k = 4$ of the multiway cut problem. We analyze several natural probability distributions over these cuts, and we develop discrete linear programs that form the basis for the computational experiments.

Our work also involves sparcs. For $k = 4$, we describe a class of probability distributions over sparcs, and specify a particular distribution for which we obtain an upper bound on the approximation ratio achieved by the algorithm when it uses the distribution as the rounding scheme. This approximation ratio improves upon the best approximation ratio for which an analytic proof has been presented.

1.4 Presentation Overview

In Chapter 2, we describe in detail the geometric embedding and the approximation algorithm for multiway cut of Călinescu et al., and the work of Karger et al. that improves the algorithm. Chapter 3 defines the cuts of the 4-simplex that we introduce to complement sparcs, and analyzes several probability distributions over these cuts. Next, Chapter 4 describes the computational experiments that we performed using the various types of cuts with which we worked. In Chapter 5, we define and analyze a probability distribution over sparcs that leads to an improved analytic approximation ratio for $k = 4$. Finally, Chapter 6 provides concluding remarks and suggestions for future work.

Chapter 2

Background

In this chapter, we describe the approximation algorithm of Călinescu, Karloff, and Rabani for the multiway cut problem. We also present the subsequent work of Karger, Klein, Stein, Thorup, and Young on probability distributions over partitions of the k-simplex that we extend in this project.

2.1 The Approximation Algorithm of Călinescu, Karloff, and Rabani

The Călinescu et al. algorithm is based on a geometric embedding of the input graph into the k-simplex $\Delta = \{x \in \mathbb{R}^k \mid x \geq 0, \sum_i x_i = 1\}$. Although the k-simplex is a $(k-1)$-dimensional convex polytope, it is convenient to use k-dimensional vectors to describe it. The k-simplex has k vertices, with the ith vertex being the point identified by the unit vector e^i, which has coordinates $(e^i)_j = 0$ for all $j \neq i$ and $(e^i)_i = 1$. To avoid ambiguity, we refer to the vertices of the input graph, the elements of V, as nodes, and we reserve the term vertex for references to vertices of the simplex.

Throughout this work, we will use one-half the L_1 norm as our distance measure. We denote the L_1 norm of a vector x by $|x|$, and the distance between two points x and y by $d(x,y)$, and thus $|x| = \sum_{i=1}^{k} |x_i|$ and $d(x,y) = (1/2)|x-y|$. The length of a line segment is the distance between its endpoints. Note that the distance between any two vertices i and j of the simplex is $(1/2)|e^i - e^j| = (1/2)\sum_{\ell=1}^{k} |(e^i)_\ell - (e^j)_\ell| = (1/2)(|(e^i)_i - (e^j)_i| + |(e^i)_j - (e^j)_j|) = (1/2)(|1-0| + |0-1|) = 1$.

2.1.1 Geometric embedding

In the embedding of the graph into the simplex, each node of the graph is mapped to a point in the simplex. Each edge is mapped to the line segment joining the points to which its endpoints map. We define the length of an embedded edge as the distance, under the one-half L_1 norm measure, between the points to which its endpoints map. The cost of an edge $(u, v) \in E$ in the graph is denoted by $c(u, v)$.

Suppose all of the nodes map to vertices of the simplex and the ith terminal maps to the ith simplex vertex. Then every embedded edge has a length of either 0 (in the case that its endpoints map to the same vertex) or 1 (in the case that its endpoints map to different vertices). A mapping of this type naturally induces a k-way cut of the graph. In this context, the set of nodes that map to a particular simplex vertex are in one connected component after the edges in the cut have been removed, and the edges in the cut are the edges whose endpoints are mapped to different vertices. Thus, the cost of the cut is the sum of the costs of the edges whose endpoints map to different vertices.

These observations provide the basis for the following nonlinear integer program, which is a formulation of the multiway cut problem. The variables in the program are vectors x^u for all nodes $u \in V$. For a node u, the variable x^u represents the point in the simplex to which u maps in the embedding.

$$\text{Minimize} \sum_{(u,v)\in E} c(u,v) d(x^u, x^v) \text{ subject to}$$

$$x^u \in \{e^i \mid i = 1, \ldots, k\} \quad \forall u \in V \tag{2.1}$$

$$x^t = e^t \quad \forall t \in T \tag{2.2}$$

The integrality constraints (2.1) require that the nodes map to vertices of the simplex. For the terminals, the program has additional constraints (2.2), which ensure that the terminals map to different vertices of the simplex. Since the distance between any two points at which nodes of the graph have been embedded is 0 if the points map to the same vertex and 1 if they map to different vertices, the integer program minimizes the sum of the costs of the edges whose endpoints map to different vertices, which is the cost of the induced k-way cut. As such, a solution to this integer program identifies the edges in a minimum-cost k-way cut of the input graph.

2.1.2 Linear programming relaxation

To further the development of their approximation algorithm for the multiway cut problem, Călinescu et al. provide a linear programming relaxation of an integer program for multiway cut. They show that their relaxed linear program is equivalent to the following relaxed formulation for multiway cut, which has one vector-valued variable x^u for each node $u \in V$.

$$\text{Minimize} \sum_{(u,v)\in E} c(u,v)d(x^u, x^v) \text{ subject to}$$

$$x^u \in \Delta \quad \forall u \in V \tag{2.3}$$

$$x^t = e^t \quad \forall t \in T \tag{2.4}$$

The integrality constraints (2.1) in the integer program have been relaxed to obtain the constraints (2.3). Under the relaxed constraints, nodes may map to any points in the simplex, not only to simplex vertices.

As in the integer program, the terminals must map to different vertices of the simplex, but the other nodes may be placed anywhere in the simplex. Clearly, a solution to this formulation in which all of the nodes in the graph map to vertices is also a solution to the integer program. This formulation allows more solutions than the integer program, however, and therefore may have a smaller optimal value than the integer program. For any feasible solution to this formulation, which is an embedding of the graph into the simplex, we refer to the value of the objective function for that solution as the *volume of the embedding*.

2.1.3 The algorithm

The approximation algorithm of Călinescu et al. for multiway cut first finds a solution to the relaxed formulation of multiway cut by linear programming. It then converts this solution to a solution to the integer program by using randomized rounding methods. The basis for the rounding scheme employed by the algorithm is a *k-way cut of the simplex*, which is a partition of the simplex into k subsets, each containing exactly one vertex of the simplex. Given a particular k-way cut of the simplex, a solution to the integer program may be obtained by placing all of the nodes that map to a particular partition of the simplex at the vertex contained in that partition. In this way, a k-way cut of the simplex induces a k-way cut of the embedded graph. An edge is in the k-way cut of the graph if its endpoints map to different partitions of the simplex.

The rounding scheme used by the algorithm selects a k-way cut of the simplex from a certain distribution. Its choice of a cut is not affected by the input graph. The k-way cut of the graph induced by the k-way cut of the simplex has an expected cost that is greater than the volume of the embedding, which is the optimal value of the linear programming relaxation, by at most a factor of $3/2 - 1/k$. As the optimal value of the linear programming relaxation is at most the cost of the minimum-cost k-way cut of the graph, the ratio of the expected cost of the cut produced by the rounding scheme to the cost of the minimum-cost k-way cut is at most $3/2 - 1/k$. Călinescu et al. show how to convert this randomized rounding scheme into a deterministic algorithm, thus providing a guarantee that their algorithm has an approximation ratio of at most $3/2 - 1/k$.

2.1.4 Integrality gap of the geometric relaxation

In the context of a relaxation formulated from a geometric embedding of a graph for a minimization problem, the *integrality gap* of the relaxation is the worst-case ratio between the value of the optimal solution to the problem and the value of the solution to the relaxation. For the Călinescu et al. relaxation, the integrality gap is the ratio, in the worst case, between the cost of the minimum-cost k-way cut of the input graph and the optimal value of the relaxed linear program. Because the integer program captures the multiway cut problem exactly, the optimal solution to the integer program is equal to the minimum k-way cut cost. Therefore, the integrality gap of the Călinescu et al. relaxation is equal to the ratio of the optimal value of the integer program to the optimal value of the linear programming relaxation. The work of Călinescu et al. shows that the integrality gap of the relaxation is at most $3/2 - 1/k$. Subsequent work by Freund and Karloff [3] demonstrates a lower bound of $8/(7 + \frac{1}{k-1})$ on the integrality gap.

2.2 A Study of the Relaxation by Karger, Klein, Stein, Thorup, and Young

The work of Karger et al. on the geometric relaxation of Călinescu et al. provides additional insight into the relaxation, as well as further results about its integrality gap. Given the algorithmic framework established by Călinescu et al., Karger et al. note that the problems of determining the integrality gap of the relaxation and finding a rounding scheme to be used in the approximation algorithm may be expressed as a geometric question. The integrality gap and the approximation ratio of the algorithm may be obtained by studying the simplex itself, without considering particular input graphs or embeddings.

2.2.1 Density

Much of the work of Karger et al. focuses on the notion of a *cutting scheme*, which is a probability distribution P over k-way cuts of the simplex. A line segment e is said to be cut by a k-way cut of the simplex if two points on the segment lie in different partitions of the simplex after the simplex is partitioned according to the k-way cut. In general, we say that a segment is cut x times by a k-way cut if the cut partitions the segment into $(x+1)$ smaller segments. The cuts that we consider are all composed of several different *slices*, where a slice is a partition of the simplex into two subsets, each containing at least one simplex vertex. We say that an individual slice cuts the segment if points on the segment lie in both subsets of the simplex induced by the slice.

For a cutting scheme P and any line segment e, Karger et al. define the *density of P on e*, denoted $\tau_k(P, e)$, to be the ratio of the expected number of times a k-way cut chosen randomly from P cuts e to the length of e. They then define the *maximum density of* P, $\tau_k(P)$, and the *minimal maximum density*, τ_k^*, as follows.

$$\tau_k(P) = \sup_e \tau_k(P, e)$$

$$\tau_k^* = \inf_P \tau_k(P)$$

Intuitively, the maximum density of P is the greatest density that P has on any segment, and the minimal maximum density is the maximum density of the cutting scheme that has the smallest maximum density. Because any line segment can be divided into two smaller segments, at least one of which has a density no less than the original segment, the segment with the maximum density will be a segment of infinitesimal length.

Since a k-way cut of the simplex converts a solution to the relaxed linear program to a solution to the integer program for multiway cut, thereby inducing a k-way cut of the input graph, any cutting scheme may be used as the randomized rounding scheme in the Călinescu et al. approximation algorithm. Karger et al. show that for any cutting scheme P and any embedded graph, the expected cost of the k-way cut of the input graph induced by a k-way cut of the simplex chosen randomly from P is greater than the volume of the embedding by at most a factor of $\tau_k(P)$. As a corollary, they conclude that the approximation ratio achieved by the algorithm when a cutting scheme P is used as the rounding scheme is at most $\tau_k(P)$. These facts imply that the maximum density $\tau_k(P)$ of any cutting scheme P is at least the integrality gap of the relaxation.

Furthermore, Karger et al. demonstrate that there exists a cutting scheme whose maximum density is equal to the integrality gap. It follows that τ_k^* is the integrality gap of the geometric relaxation.

2.2.2 Alignment

The work of Karger et al. demonstrates that the property of a cutting scheme that determines its performance as the rounding scheme for the approximation algorithm is the maximum density the cutting scheme has on any line segment. Călinescu et al. showed that it is only necessary to consider segments in certain orientations. A line segment in the simplex is *i, j-aligned* if it is parallel to the edge whose endpoints are vertices i and j of Δ. Because vertices i and j are e^i and e^j, the absolute value of the difference in coordinate ℓ of the vertices is $|(e^i)_\ell - (e^j)_\ell| = 1$ for $\ell \in \{i, j\}$ and $|(e^i)_\ell - (e^j)_\ell| = 0$ for $\ell \notin \{i, j\}$. Therefore, an i, j-aligned segment (x, y) satisfies $|x_\ell - y_\ell| = d(x, y)$ for $\ell \in \{i, j\}$ and $|x_\ell - y_\ell| = 0$ for $\ell \notin \{i, j\}$. We say that a segment is *aligned* if it is i, j-aligned for some pair of vertices i and j.

Călinescu et al. noted that the endpoints of any segment can be connected by a piecewise linear path whose total length is equal to the length of the original segment and whose linear pieces are aligned segments. This follows from the linear relationship between length and the L_1 norm, and the fact that, under the distance measure, an aligned segment is the shortest line segment between points that may be connected with an aligned segment. Figure 2-1 shows an example of the endpoints of a segment being connected by a piecewise linear path consisting of aligned pieces. A particular segment in the simplex is cut if and only if some segment on the piecewise linear path of aligned segments between its endpoints is cut. For any embedding of a graph, the Călinescu et al. approximation algorithm applies this transformation to each segment between two embedded points, without affecting the value of the optimal solution to the relaxed linear program. The implication of this transformation for the concept of density introduced by Karger et al. is that in order to determine the maximum density of a cutting scheme, one must consider only aligned segments. As a result, without loss of generality one may consider only embeddings of the graph into the simplex in which all edges are aligned.

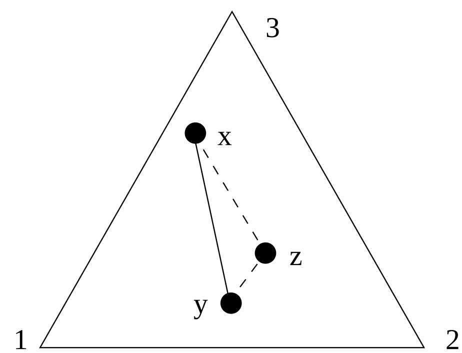


Figure 2-1: The points $x = (1/4, 1/8, 5/8)$ and $y = (7/16, 7/16, 1/8)$ in the 3-simplex may be connected in several ways. While the segment (x, y) is not aligned, it is possible to connect x and y by a piecewise linear path through a third point, $z = (1/4, 7/16, 5/16)$, such that (x, z) is 2, 3-aligned and (z, y) is 1, 3-aligned. The length of (x, y) is $(1/2)(|1/4 - 7/16| + |1/8 - 7/16| + |5/8 - 1/8|) = (1/2)(3/16 + 5/16 + 1/2) = 1/2$. As the sum of the lengths of (x, z) and (z, y) is $(1/2)(|1/4 - 1/4| + |1/8 - 7/16| + |5/8 - 5/16|) + (1/2)(|1/4 - 7/16| + |7/16 - 7/16| + |5/16 - 1/8|) = (1/2)(0 + 5/16 + 5/16) + (1/2)(3/16 + 0 + 3/16) = 5/16 + 3/16 = 1/2$, the total length of the piecewise linear path consisting of (x, z) and (z, y) is equal to the length of the segment (x, y).

2.2.3 Side-parallel cuts

Karger et al. performed a series of computational experiments with the goal of identifying cutting schemes that lead to small approximation ratios. Their experiments involved a particular type of k-way cut of the simplex that they refer to as a *side-parallel cut*, or *sparc*. A sparc is composed of a sequence of *side-parallel slices*, where an individual slice is defined as $\Delta_{x_i=\rho} = \{x \in \Delta \mid x_i = \rho\}$. A slice $\Delta_{x_i=\rho}$ is a hyperplane that is parallel to the face opposite terminal i and is at a distance ρ from that face. It partitions the simplex into two regions, one of which is defined as $\Delta_{x_i \geq \rho} = \{x \in \Delta \mid x_i \geq \rho\}$ and may be considered to be the "corner" of the simplex containing terminal i.

A side-parallel cut is defined by Karger et al. as the following procedure, which yields a k-way cut of the simplex.

1. Choose a permutation σ of the simplex vertices.

2. Process the vertices in the order specified by σ. For each vertex i (except possibly the last), choose a slice distance $\rho_i \in [0, 1]$.

3. When a vertex i is processed, assign to vertex i all the points in $\Delta_{x_i \geq \rho}$ that have not already been assigned to a previous terminal. We say that terminal i, the node in the graph mapped to vertex i, *captures* the points assigned to it, and *cuts* an edge if it captures some of the points in the edge, but not the entire edge.

Each of the hyperplane slices $\Delta_{x_i=\rho}$ used in a sparc assigns a region of the simplex to a particular vertex i. Once a sparc has applied $k-1$ slices to the simplex, there are $k-1$ regions of the simplex that are associated with vertices. At that point, it is natural to assign all of the remaining unassigned points in the simplex to the final vertex in the permutation. In this way, $k-1$ side-parallel slices produce a k-way cut of the simplex.

Consider an i,j-aligned segment (x, y). Terminal ℓ cuts the segment with a side-parallel slice $\Delta_{x_\ell=\rho}$ only if ρ lies between x_ℓ and y_ℓ. Because $|x_\ell - y_\ell| = 0$ for $\ell \notin \{i, j\}$, terminal ℓ cannot cut the segment if $\ell \notin \{i, j\}$. As a result, an i,j-aligned segment is cut by a sparc only if it is cut by terminal i or terminal j.

Karger et al. show that there is an optimal sparc cutting scheme in which the slice distances are chosen from some probability distribution, and then the simplex is partitioned using the slices by processing the vertices in the order specified by a permutation selected uniformly at random from the set of all permutations of the vertices. They note that an analogous "order-independence" property also applies to the best possible cutting scheme, which may not be a distribution over sparcs. As such, the approximation ratio of the algorithm cannot be improved by using a particular ordering of the terminals to partition the simplex in lieu of a uniformly-random ordering.

The randomized rounding scheme given by Călinescu et al. for their approximation algorithm may be expressed as a probability distribution over sparcs. Their rounding scheme selects a value ρ uniformly at random from $[0, 1]$. It randomly chooses one of two candidate permutations of the vertices, and applies the side-parallel slices using slice distances $\rho_i = \rho$ for all i. Călinescu et al. show that this cutting scheme has a maximum density of $3/2 - 1/k$.

2.2.4 Computational experiments

To find distributions over k-way cuts of the simplex that minimize the maximum density of a segment in the simplex, Karger et al. formulated an infinite-dimensional linear program in which there is one variable for each cut of the simplex, which may be interpreted as the probability that the cut is selected in a rounding scheme. For every aligned and infinitesimal line segment in the simplex, the linear program has a constraint that captures the density on the segment induced by the probability distribution specified by the values of the variables. The solution to the program is an assignment of probabilities to the various k-way cuts of the simplex, which may be considered to be a cutting scheme. While an infinite-dimensional linear program cannot be solved in practice using computational techniques, Karger et al. were able to solve discrete versions of the linear program and use the solutions to improve the approximation ratio of the Călinescu et al. algorithm.

For the general case of k terminals, Karger et al. used sparcs as the only type of k-way cut present in the linear program. Based on their observation that there is an optimal sparc cutting scheme in which the vertices are permuted uniformly at random before the slices are applied, they worked only with sparcs that choose the permutation of the vertices uniformly at random from the set of possible permutations. To obtain a discrete linear program, they first fixed an integer *grid size* N. For a particular vertex i, the set $[0, 1]$ of possible side-parallel slice distances is divided into N disjoint *slabs*, where slab j is the set $[j/N, (j+1)/N]$ for $j = 0, \ldots, N-1$.

A *discrete sparc* is identified by a sequence $(u_1, \ldots, u_{k-1})$ of slabs, where u_i is an integer in the range $[0, N-1]$, which specifies the slab in which the ith slice lies. The conditional distribution of a slice distance is uniform over the slab in which the slice lies. That is, given that the ith slice is in slab u_i, the slice distance of the ith slice is chosen uniformly at random from $[u_i/N, (u_i+1)/N]$. The discrete linear program has one variable for each discrete sparc. Given a particular solution to the linear program, which is a probability distribution over discrete sparcs, a cutting scheme may be obtained by using the conditional distributions of the slice distances within the slabs to compute a probability distribution over continuous sparcs.

In order to limit the number of constraints in the linear program to a finite number, Karger et al. made use of the slabs defined as a partition of the set of possible side-parallel slice distances. A *cell* of the simplex is described by a vector $(q_1, \ldots, q_k)$, where q_i is an integer in the range $[0, N-1]$ that specifies the slab along the ith coordinate axis that contains the cell. The cell consists of the points in the set $\{x \mid q_i/N \leq x_i \leq (q_i+1)/N, i = 1, \ldots, k\}$. Figure 2-2 shows the cells in the simplex for $k = 3$ and $N = 3$. For each cell, the discrete linear program has one constraint.

The purpose of each constraint is to provide an upper bound on the density of any segment with a certain alignment in the cell. If the cells are sufficiently small, then, under any continuous distribution over k-way cuts, all segments with a particular alignment in a cell would have approximately the same density, and so one would expect that the upper bound for that cell would be nearly tight. Karger et al. compute the density bound for segments with a certain alignment a in a cell q by applying the total probability theorem, conditioning on the choice of the discrete sparc. Let U be the set of discrete sparcs in the program, H_u be the event that a discrete sparc u is chosen, and $\psi(u, q, a)$ be the upper bound on the density of the discrete sparc u

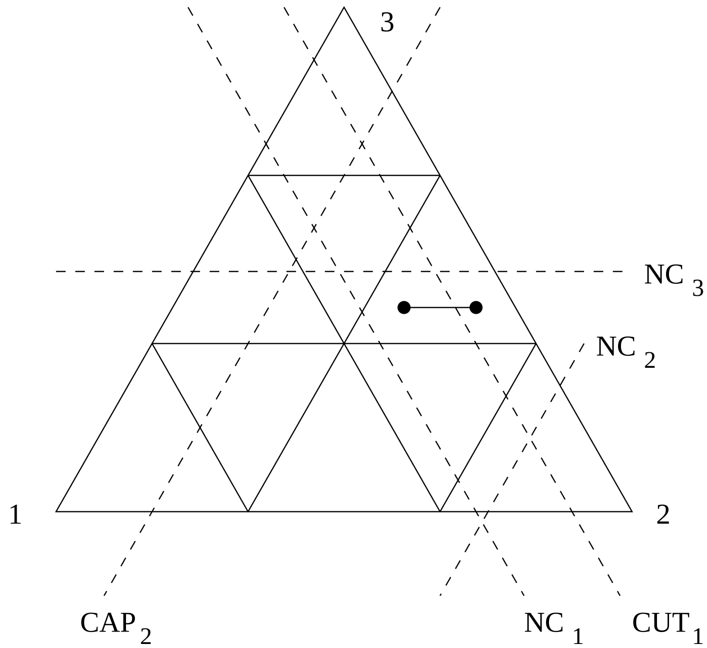


Figure 2-2: Discrete cells of the simplex formed by the intersection of the slabs for $k = 3$ and $N = 3$. The side-parallel slices illustrate the different locations relative to a segment where a slice may fall. Slices labelled NC do not cut or capture the segment. The slice labelled CUT cuts the segment, while the slice labelled CAP captures it, but does not cut it. The subscript in the label for each slice indicates the vertex of the simplex that is captured by the slice.

on any segment with alignment a in the cell q. Denote by C the number of times a segment of length d with alignment a in the cell is cut by a sparc chosen randomly according to the distribution induced by the discrete sparcs. The expected number of times the segment is cut is $E[C] = \sum_{u \in U} \Pr[H_u] E[C \mid H_u]$. Thus, an upper bound on the density of the distribution induced by the discrete sparcs on any segment with alignment a in the cell q may be computed as $\sum_{u \in U} \Pr[H_u] \psi(u, q, a)$. Note that $\Pr[H_u]$ is the value of variable for the discrete sparc u in the program, while $\psi(u, q, a)$ is an upper bound on the density of the discrete sparc u on any segment with alignment a in the cell q that is computed for each choice of discrete sparc u, cell q, and alignment a.

In the constraint, Karger et al. require that the density bound for each alignment a and cell q be less than the value of a variable λ. The objective of the linear program is to minimize λ. Since λ is required by the constraints to be an upper bound on the density of any segment in any cell of the simplex, it will be the maximum density of the cutting scheme when the linear program is solved. By minimizing λ, the linear program, subject to the discrete formulation of the problem, finds the cutting scheme with the minimal maximum density.

For a particular discrete sparc u, the upper bound $\psi(u, q, a)$ on the density of u on any segment with a certain alignment a in a cell q is computed as follows. Each of the $k!$ permutations of the vertices is considered in turn. Because each permutation is equally likely to be selected by the sparc, the final density bound is the average of the density bounds for the different permutations. For each permutation, the vertices are processed in the order specified by the permutation, and the ℓth slice distance u_ℓ of the discrete cut is applied to the ℓth vertex in the permutation, $\sigma(\ell)$. The expected number of cuts of a segment in the cell by the ℓth slice is determined by the relative values of the slice distance u_ℓ and the slab along coordinate $\sigma(\ell)$ in which the cell lies, $q_{\sigma(\ell)}$. If $u_\ell < q_{\sigma(\ell)}$, as shown in Figure 2-2 for the slice CAP_2, then terminal $\sigma(\ell)$ captures all the segments in the cell because the cell lies in the corner containing terminal $\sigma(\ell)$, and so it does not cut any segments in the cell. In the case that $u_\ell > q_{\sigma(\ell)}$, as illustrated by the slice NC_2 in Figure 2-2, the slice does not pass through the cell, nor does it capture any segment in the cell.

When $u_\ell = q_{\sigma(\ell)}$, the slice passes through the cell, so it may cut some segments in the cell. In Figure 2-2, the slices NC_1 and CUT_1 both pass through the cell containing the segment shown, but only the slice CUT_1 cuts the segment. An i, j-aligned segment (x, y) can be cut only if $\sigma(\ell) \in \{i, j\}$. Note that in Figure 2-2, the segment shown is $1, 2$-aligned, and therefore is parallel to the slice NC_3 for terminal 3 and cannot be cut by it. Because the conditional distribution of the slice distance within the slab identified by u_ℓ is uniform, and $|x_i - y_i| = |x_j - y_j| = d(x, y)$, the expected number of times the segment (x, y) is cut is $d(x, y)/(1/N) = Nd(x, y)$, and thus the density on the segment is $(Nd(x, y))/d(x, y) = N$. As this calculation applies to any segment in the cell, the density of the slice on any segment in the cell, conditional on the choice of the discrete sparc, is at most N. Thus, the density bound for the cell is N

if exactly one of the slices for terminals i and j pass through the cell, and $2N$ if both slices pass through it.

Karger et al. exploited the symmetry of the random permutations to limit the number of alignments explicitly examined during the process of generating the linear program. Since all of the sparcs that they consider permute the vertices uniformly at random, two segments that have the same endpoints under permutation of coordinates will have the same densities, and thus only one of them must be present in the program. The linear program includes only cells $(q_1, \ldots, q_k)$ in which the coordinates q_i are in nondecreasing order, and the segments within the cells are assumed to be $1, 2$-aligned.

The resulting discrete linear program has the following form. Denote the set of discrete sparcs by U, and the set of cells by Q. Let p_u be the variable for a discrete sparc u in the program. For a discrete sparc u and a cell q, let $\psi(u, q)$ denote the computed upper bound on the density of any $1, 2$-aligned segment within cell q, conditional on the event that the sparc used to partition the simplex is chosen from the continuous distribution induced by the discrete sparc u.

Minimize λ subject to

$$p_u \geq 0 \qquad \forall u \in U \tag{2.5}$$

$$\textstyle\sum_{u \in U} p_u = 1 \tag{2.6}$$

$$\textstyle\sum_{u \in U} \psi(u, q) p_u \leq \lambda \quad \forall q \in Q \tag{2.7}$$

As the variables p_u represent the probabilities $\Pr[H_u]$ of choosing the various discrete sparcs, the constraints (2.5) and (2.6) require that they form a probability distribution. The constraints (2.7) ensure that the objective value λ is the maximum density of the cutting scheme.

The dual of the discrete linear program conforms to the following structure. It has one variable c_q for each cell q in the simplex.

Maximize μ subject to

$$c_q \geq 0 \qquad \forall q \in Q \tag{2.8}$$

$$\textstyle\sum_{q \in Q} c_q = 1 \tag{2.9}$$

$$\textstyle\sum_{q \in Q} \psi(u, q) c_q \geq \mu \quad \forall u \in U \tag{2.10}$$

This discrete program is indicative of the dual of the infinite-dimensional primal linear program in which the variables are cuts of the simplex. In the dual program, the variables are line segments in the simplex, and there is one constraint for each cut of the simplex. The value of a variable may be considered to be a cost assigned to the line segment corresponding to that variable. If we consider constructing an embedded graph in which all segments that have positive costs are present, we may interpret a solution to the dual program as a particular embedded graph. For any k-way cut u of the simplex, the quantity $\sum_{q \in Q} \psi(u, q) c_q$ may be interpreted as the cost of the k-way cut. Because the constraints require that this quantity be at least μ for every cut, and the linear program maximizes μ, the program attempts to ensure that the minimum cost of a k-way cut over all cuts in the program is as large as possible.

In the primal linear program, the objective is to assign probability to the various cuts of the simplex such that the maximum density induced by the cuts on any line segment is minimized. A solution to the program is a probability distribution over cuts of the simplex, which is a cutting scheme. In the dual program, the objective is to assign a total of one unit of cost to the line segments such that the cost of the minimum-cost k-way cut of the simplex is maximized. Effectively, an optimal solution to the primal program is an optimal cutting scheme, while an optimal solution to the dual program is an embedded graph whose edges have been assigned a fixed amount of total cost on which any cut will have a high cost.

2.2.5 Results

For the case of $k = 3$ terminals, Karger et al. exploited the planarity of the 3-simplex to develop primal and dual linear programs that were specific to the 3-terminal case. The dual solutions led them to construct a particular embedded graph that they used to prove a lower bound of $\tau_3^* \geq 12/11$ on the integrality gap of the relaxation. They then used the general form of the dual solution to determine the necessary structure of any optimal primal solution. They present a cutting scheme that has a maximum density of 12/11, thus demonstrating an upper bound of $\tau_3^* \leq 12/11$ on the integrality gap. From the matching lower and upper bounds, they conclude that the integrality gap in the three-terminal case is precisely $\tau_3^* = 12/11$.

In the general case of $k > 4$ terminals, Karger et al. used observations from their computational experiments to specify improved cutting schemes. The cutting scheme that they give for k terminals is a probability distribution over sparcs, and involves *corner cuts*. Since a side-parallel slice $\Delta_{x_i=\rho_i}$ is a hyperplane parallel to the face opposite vertex i of the simplex that is at a distance ρ_i from that face, the larger the value of the slice distance ρ_i, the closer the slice is to the corner of the simplex containing terminal i. A corner cut is a sparc in which the slice distances ρ_i are all selected from some range $[c, 1]$, where c is a parameter that determines the size of the corner. The cutting scheme chooses a corner cut with a certain probability, and otherwise selects each slice distance ρ_i independently and uniformly from $[0, c]$. Karger et al. show that their cutting scheme for general k achieves an approximation ratio of at most 1.3438 for all k, and that the integrality gap is therefore at most 1.3438 for all k.

Karger et al. noted that when k is small, it is possible to refine the analysis that they used to prove the upper bound on the maximum density of their cutting scheme for general k. This approach allowed them to improve the approximation ratios produced by the cutting scheme for particular values of k. For $k \in \{3, 4, 5, 6, 7, 8, 9, 10, 12, 20, 35\}$, they tuned the parameters of the cutting scheme for each value of k, with the goal of minimizing the approximation ratio. The resulting approximation ratios are the smallest ones that they report for $k \geq 6$. In the cases of $k = 4$ and $k = 5$ terminals, respectively, Karger et al. give analytic upper bounds of 1.189 and 1.223 on the approximation ratio achieved by the cutting scheme.

The discrete primal linear programs that Karger et al. solved for the general case of k terminals included upper bounds on the densities induced by cuts on segments in the constraints. These upper bounds ensure that the optimal value of the discrete linear program is an upper bound on the maximum density of the cutting scheme specified by the solution to the linear program. Thus, the solutions to the discrete linear programs serve as computational proofs of upper bounds on the maximum densities of the cutting schemes. Moreover, because the primal linear program solutions are distributions over discrete sparcs, and each discrete sparc induces a distribution over continuous sparcs, it would be possible to verify analytically the maximum densities of the cutting schemes corresponding to the solutions by a case analysis on each cell of the simplex. For the cases of $k = 4$ and $k = 5$ terminals, the smallest approximation ratios obtained by Karger et al. come from the solutions to the discrete linear programs, as each solution to the primal program specifies a cutting scheme whose maximum density is an upper bound on the approximation ratio of the Călinescu et al. algorithm when the cutting scheme is used as the rounding scheme in the algorithm. In particular, for $k = 4$, Karger et al. report an upper bound of 1.1539 on the approximation ratio achieved by the optimal cutting scheme.

Chapter 3

Additional Cuts of the 4-Simplex

Our extensions to the computational experiments of Karger et al. are based on including alternatives to sparcs in the set of cuts of the simplex used in the linear programs. The cuts that we introduce are defined for the 4-simplex in particular, and thus our experiments focus on the case of $k = 4$ terminals. In this chapter, we describe the cuts that we consider as alternatives to sparcs in the experiments. We also analyze two natural cutting schemes that use these cuts in which the slice distances are chosen uniformly at random from the range of possible values.

A side-parallel slice isolates a single vertex of the simplex from the other vertices by dividing the simplex into two regions, one of which is the corner containing the vertex. As a sparc is comprised of side-parallel slices, it partitions the simplex by isolating vertices one at a time. The cuts that we introduce in addition to sparcs first isolate two pairs of vertices from each other, then separate each pair of vertices. In general, we refer to any cut that partitions the 4-simplex by isolating pairs of vertices and then separating the pairs as a *pair-isolating cut.* We define two different types of pair-isolating cuts that differ in the way in which they separate the pairs of vertices.

In the analysis that we present, we demonstrate that the two natural cutting schemes that use only pair-isolating cuts both have maximum densities of $4/3$. The analogous sparc cutting scheme in which the slice distances are chosen uniformly at random from the range of possible values has a maximum density of $5/4$, and therefore the results of the analysis suggest that pair-isolating cuts are not as effective as sparcs when used as the sole type of cut in the rounding scheme of the Călinescu et al. approximation algorithm for the multiway cut problem. The results of the computational experiments that we describe in Chapter 4, however, provide evidence that combining pair-isolating cuts with sparcs in a cutting scheme may yield a better approximation ratio than using either type of cut in isolation.

3.1 Pair-Isolating Slices

The symmetry of the k-simplex ensures that using k-way cuts of the simplex that are not symmetric with respect to all vertices in a cutting scheme will not be effective. Sparcs that use random permutations of the vertices are attractive cuts because they exhibit symmetry. For the 4-simplex, a regular tetrahedron, there is another type of symmetry that we attempt to exploit with our additional cuts. The key observation is that the 4-simplex may be divided into two regions that have the same shape, each containing two vertices. This symmetry between the two regions containing pairs of vertices provides a starting point for our cuts.

A sparc partitions the simplex into k subsets by isolating vertices into their own regions one at a time. The cuts that we introduce for the tetrahedron first apply a slice that we refer to as a *pair-isolating slice* to isolate one pair of vertices from the other pair. A pair-isolating slice divides the simplex into two regions, each of which contains two vertices. To obtain a four-way cut of the simplex, we then use one slice in each of these two regions to separate the two vertices in the region.

To ensure a symmetry between the two regions of the simplex produced by the pair-isolating slice, we use hyperplanes that are parallel to the two edges of the simplex connecting the pairs of vertices that they isolate. These hyperplanes are normal to a particular set of vectors. Consider the problem of choosing a slice that isolates vertices s_1 and s_2 from vertices t_1 and t_2, where s_1, s_2, t_1, t_2 is a permutation of $1, 2, 3, 4$. Let m_s denote the midpoint of the edge that connects vertices s_1 and s_2, and let m_t denote the midpoint of the edge that joins t_1 and t_2. We will isolate the pairs of vertices by applying as our slice a hyperplane that is normal to the line segment (m_s, m_t). Such a hyperplane is parallel to the edge that joins vertices s_1 and s_2, and to the edge that joins vertices t_1 and t_2. It partitions the simplex into two regions of the same shape, though the regions will have different sizes that depend on where the slice hyperplane intersects the line segment. Figure 3-1 shows an example of a hyperplane that isolates vertices s_1, s_2 from vertices t_1, t_2.

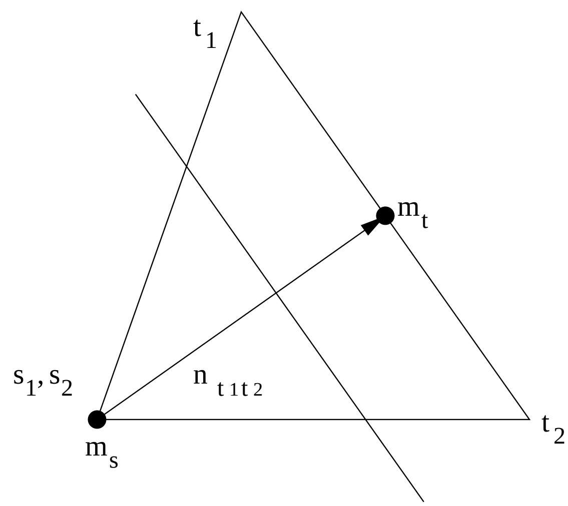


Figure 3-1: A pair-isolating slice that isolates the pair of vertices s_1, s_2 from the pair t_1, t_2, using $n_{t_1 t_2}$ as the basis vector. Note that in this diagram, the edge connecting vertices s_1 and s_2, of which m_s is the midpoint, is perpendicular to the plane of the page.

As in the sparc case, we choose the slice distance for the hyperplane from the range $[0, 1]$. Either m_s or m_t may be considered as the point at which a hyperplane for slice distance 0 intersects the normal line segment, with the other point the intersection of the segment and the hyperplane for distance 1. If m_s is chosen to correspond to distance 0, then we refer to the vector $m_t - m_s$ as the *basis vector* of the pair-isolating slice, and we denote it by $n_{t_1 t_2} = m_t - m_s$. Note that choosing vertices s_1 and s_2 to be in the same pair leads to the same group of two pairs as choosing t_1 and t_2 to be in a single pair, and as a result there are $\binom{4}{2}/2 = 3$ distinct choices of two pairs of four vertices. For each choice of pairs, there are two choices for the point that corresponds to slice distance 0, and so there are $3 \times 2 = 6$ distinct basis vectors that may be used for the application of the pair-isolating slice. There is one basis vector n_{ij} for each of the $\binom{4}{2} = 6$ pairs of distinct vertices i, j.

The point m_s, the midpoint of the edge that connects vertices s_1 and s_2, has coordinates $(m_s)_{s_1} = (m_s)_{s_2} = 1/2$ and $(m_s)_{t_1} = (m_s)_{t_2} = 0$. Similarly, the coordinate values for m_t are $(m_t)_{t_1} = (m_t)_{t_2} = 1/2$ and $(m_t)_{s_1} = (m_t)_{s_2} = 0$. Therefore, the basis vector $n_{t_1t_2} = m_t - m_s$ has coordinate values $(n_{t_1t_2})_{t_1} = (n_{t_1t_2})_{t_2} = 1/2$ and $(n_{t_1t_2})_{s_1} = (n_{t_1t_2})_{s_2} = -1/2$.

A hyperplane normal to the vector $n_{t_1t_2}$ may be described by $(1/2)(x_{t_1} + x_{t_2}) - (1/2)(x_{s_1} + x_{s_2}) = C$ for some constant C. The hyperplane normal to $n_{t_1t_2}$ that passes through m_s satisfies $C = -1/2$, whereas the hyperplane that goes through m_t corresponds to $C = 1/2$. In general, the constant C and the slice distance ρ are related by $C = \rho - 1/2$. Using the fact that points in the simplex satisfy $(x_{s_1} + x_{s_2}) + (x_{t_1} + x_{t_2}) = 1$, as well as the relationship $\rho = C + 1/2$, we may substitute for $(x_{s_1} + x_{s_2})$ and C in the equation for the hyperplane to obtain $x_{t_1} + x_{t_2} = \rho$ as a description of the points in the simplex that lie on the hyperplane for slice distance ρ. Accordingly, we define $\Delta_{x_i+x_j=\rho} = \{x \in \Delta \mid x_i + x_j = \rho\}$. To describe the two regions on either side of the pair-isolating slice, we define $\Delta_{x_i+x_j<\rho} = \{x \in \Delta \mid x_i + x_j < \rho\}$ and $\Delta_{x_i+x_j\geq\rho} = \{x \in \Delta \mid x_i + x_j \geq \rho\}$.

3.2 Separating Pairs of Vertices

A pair-isolating slice $\Delta_{x_i+x_j=\rho}$ partitions the simplex into two regions. We refer to $\Delta_{x_i+x_j<\rho}$ as the *source region*, and to $\Delta_{x_i+x_j\geq\rho}$ as the *sink region*. Each region contains two vertices. In order to complete a 4-way cut of the simplex, it is necessary at this point to partition each of the two regions so that there is exactly one vertex in each partiton of the simplex. Any method of dividing a region containing a pair of vertices such that the vertices are separated into different subsets may be referred to as a *pair-separating slice*. We consider two types of pair-separating slices.

3.2.1 Side-parallel slices

One way to separate a pair of vertices is to use a side-parallel slice. In this case, one of the two vertices in the region is selected. We refer to this vertex as the *reference vertex* for the side-parallel slice. A slice parallel to the face opposite to the reference vertex is used to divide the region into two subsets. If vertex i is the reference vertex, then the side-parallel slice may be described as $\Delta_{x_i=\rho}$ for some slice distance ρ. All of the points in the region and in $\Delta_{x_i \geq \rho}$ are assigned to vertex i, and the points that are in the region but not in $\Delta_{x_i \geq \rho}$ are assigned to the other vertex in the region.

The side-parallel slice intersects the pair-isolating slice in a line. For the purpose of determining whether a segment is cut by the 4-way cut, we may consider the side-parallel slice to terminate at this line. That is, the slice that we use to separate the vertices in one region will not cut any segments in the other region, as we do not partition the other region based on the slice that separates the vertices. Effectively, the source or the sink region captures segments so that pair-separating slices in the other region will not cut those segments. Note, however, that because each pair-separating slice is used to divide only one region and there is only one such slice per region, the capture of segments by a pair-separating slice does not provide a benefit in the area of a reduction in the number of cuts on segments by other slices.

We define a *pair-isolating side-parallel cut*, abbreviated pair-side cut, of the 4-simplex as follows.

1. Choose a basis vector n_{ij} from the set of six possibilities.

2. Choose a slice distance $\rho_p \in [0, 1]$ for the pair-isolating slice.

3. Select a slice distance $\rho_s \in [0, 1]$ for the source region. Choose a vertex i_s from the set of two vertices in the source region. Assign to vertex i_s all the points in $\Delta_{x_i+x_j<\rho_p} \cap \Delta_{x_{i_s} \geq \rho_s}$, and assign to the other vertex in the source region all the remaining points in the source region.

4. Select a slice distance $\rho_t \in [0, 1]$ for the sink region. Choose a vertex i_t from the set of two vertices in the source region. Assign to vertex i_t all the points

in $\Delta_{x_i+x_j\geq\rho_p} \cap \Delta_{x_{i_t}\geq\rho_t}$, and assign to the other vertex in the sink region all the remaining points in the sink region.

Figure 3-2 shows how the sink region is partitioned by a pair-isolating side-parallel cut.

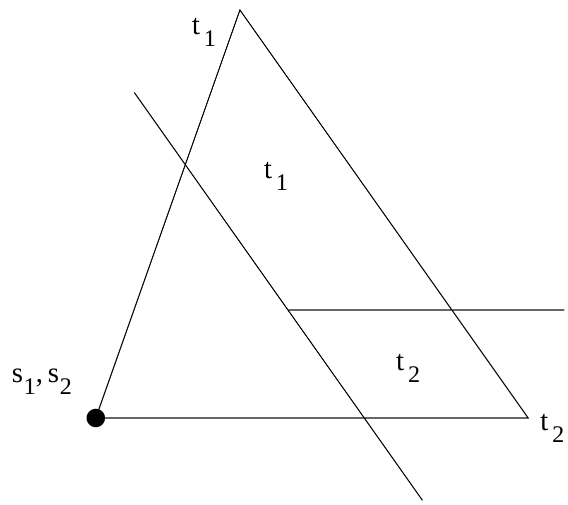


Figure 3-2: The partition of the sink region produced by a pair-side cut. For the source region, the side-parallel slice does not appear because it is not perpendicular to the plane of the page and therefore would be difficult to visualize. The reference vertex for the side-parallel slice in the sink region is vertex t_1, and the two parts of the sink region induced by the side-parallel slice are labelled with the vertices to which they are assigned.

3.2.2 Edge-perpendicular slices

When a side-parallel slice is used to partition a region produced by a pair-isolating slice, the resulting two regions are polytopes that are not similar, as Figure 3-2 illustrates. An alternative to a side-parallel slice for the separation of a pair of vertices is a slice that is symmetric with respect to the region containing the two vertices to be separated. Using such a slice would result in a symmetry among the regions containing the various vertices of the simplex.

A hyperplane normal to the edge that joins two vertices divides a region on one side of a pair-isolating cut into two regions of the same shape. As a result, we consider these types of hyperplanes as our candidates for separating pairs of vertices. Fix two vertices i and j, and consider the vector $e^j - e^i$. This vector may be considered to be a directed representation of the edge from i to j, and has coordinate values $(e^j - e^i)_j = 1$, $(e^j - e^i)_i = -1$, and $(e^j - e^i)_\ell = 0$ for $\ell \notin \{i, j\}$. A hyperplane normal to this vector satisfies $x_j - x_i = C$ for some constant C. Such a hyperplane partitions a subset of the simplex resulting from a pair-isolating cut into two regions of the same shape.

The hyperplane normal to $e^j - e^i$ that passes through vertex i is the one for $C = -1$. Similarly, the hyperplane normal to the vector that contains vertex j corresponds to $C = 1$. As such, the hyperplanes that we consider as candidates for separating pairs of vertices are those with constants C in the range $[-1, 1]$, which are the hyperplanes that pass through and therefore divide the simplex. For these slices, we allow the slice distance ρ to be drawn from the range $[-1, 1]$, so that the slice distance is equal to the value of C for the hyperplane that it specifies.

We define an *edge-perpendicular slice* as $\Delta_{x_j - x_i = \rho} = \{x \in \Delta \mid x_j - x_i = \rho\}$. We refer to the vector $e^j - e^i$ as the basis vector of the edge-perpendicular slice. As with the other types of slices, an edge-perpendicular slice partitions the simplex into two regions, $\Delta_{x_j - x_i < \rho} = \{x \in \Delta \mid x_j - x_i < \rho\}$ and $\Delta_{x_j - x_i \geq \rho} = \{x \in \Delta \mid x_j - x_i \geq \rho\}$. When a region is divided using an edge-perpendicular slice $\Delta_{x_j - x_i = \rho}$, the points in the region in $\Delta_{x_j - x_i < \rho}$ are assigned to vertex i, and in the points in the region in $\Delta_{x_j - x_i \geq \rho}$ are assigned to vertex j.

Analogous to a pair-isolating side-parallel cut, we define a *pair-isolating edge-perpendicular cut* of the 4-simplex, abbreviated pair-edge cut, as the following procedure.

1. Choose a basis vector $n_{t_1 t_2}$ for the pair-isolating slice from the set of six possibilities. The sink region will contain the vertices t_1 and t_2, while the source region will contain two vertices s_1 and s_2.

2. Choose a slice distance $\rho_p \in [0, 1]$ for the pair-isolating slice.

3. Select a slice distance $\rho_s \in [-1, 1]$ for the source region. Choose one of the

two possible basis vectors for the edge-perpendicular slice by identifying one of the vertices s_1, s_2 in the source region as the vertex s_ℓ corresponding to slice distance $\rho = -1$, and the other vertex as the vertex s_h corresponding to slice distance $\rho = 1$. Assign to vertex s_ℓ all the points in $\Delta_{x_{t_1}+x_{t_2}<\rho_p} \cap \Delta_{x_{s_h}-x_{s_\ell}<\rho_s}$, and to vertex s_h all the points in $\Delta_{x_{t_1}+x_{t_2}<\rho_p} \cap \Delta_{x_{s_h}-x_{s_\ell}\geq\rho_s}$.

4. Select a slice distance $\rho_t \in [-1, 1]$ for the sink region. Choose one of the two possible basis vectors for the edge-perpendicular slice by identifying one of the vertices t_1, t_2 in the sink region as the vertex t_ℓ corresponding to slice distance $\rho = -1$, and the other vertex as the vertex t_h corresponding to slice distance $\rho = 1$. Assign to vertex t_ℓ all the points in $\Delta_{x_{t_1}+x_{t_2}\geq\rho_p} \cap \Delta_{x_{t_h}-x_{t_\ell}<\rho_t}$, and to vertex t_h all the points in $\Delta_{x_{t_1}+x_{t_2}\geq\rho_p} \cap \Delta_{x_{t_h}-x_{t_\ell}\geq\rho_t}$.

Figure 3-3 displays the separation of the sink region by a pair-isolating edge-perpendicular cut.

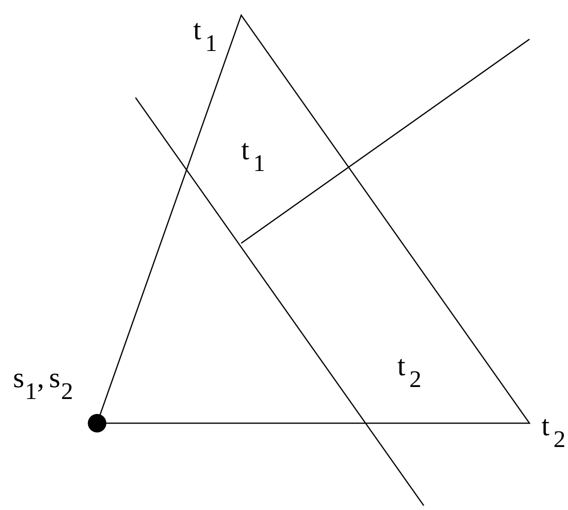


Figure 3-3: The partition of the sink region produced by a pair-edge cut. Note that in this diagram, the edge-perpendicular slice in the source region is parallel to the plane of the page, and so the partition of the source region is not visible. The two parts of the sink region induced by the edge-perpendicular slice are labelled with the vertices to which they are assigned.

3.3 Analysis of Uniform Distributions Over Pair-Isolating Cuts

To illustrate some of the properties of the two types of pair-isolating cuts that we have defined, we analyze cutting schemes involving each type. We focus on natural cutting schemes in which each slice distance is drawn uniformly at random from the range of possible slice distances, independently of the other slice distances.

Because of the symmetry of the simplex and the order-independence property shown by Karger et al. for the optimal cutting scheme, we consider cutting schemes that are symmetric with respect to all coordinates. To ensure that the cutting schemes over pair-isolating cuts are symmetric with respect to all coordinates, the choice of the basis vector for the pair-isolating slice is made uniformly at random from the set of six possibilities. For pair-side cuts, in each region produced by the pair-isolating cut, the reference vertex is chosen uniformly at random from the set of two vertices in the region. In the case of pair-edge cuts, the basis vector for the the edge-perpendicular slice in each region is chosen uniformly at random from the two possibilities.

When a cutting scheme is symmetric with respect to all coordinates, any aligned segment may be mapped to a corresponding $1,2$-aligned segment for which the density of the cutting scheme will be equal on the two segments. As a result, to compute the maximum density of a cutting scheme that is symmetric with respect to all coordinates, it is sufficient to compute the maximum density of the scheme on $1,2$-aligned segments. Therefore, we consider only $1,2$-aligned segments in our analysis when we study the maximum densities of these cutting schemes.

We determine the maximum density of a cutting scheme by computing the density of the scheme on a $1,2$-aligned segment s of length d in the simplex. As the density of a cutting scheme on a segment is defined as the ratio between the expected number of times a random cut from the cutting scheme cuts the segment and the length of the segment, we compute the expected number of times the segment is cut when a cut is chosen according to the specified distribution, and divide that quantity by d.

Let the endpoints of the segment be $(s_1 + d, s_2, s_3, s_4)$ and $(s_1, s_2 + d, s_3, s_4)$. For any particular segment s, the quantities s_1, s_2, s_3, s_4 and d are fixed constants. Because the segment is in the simplex, the coordinates of its endpoints and its length satisfy the relationship $s_1 + s_2 + s_3 + s_4 + d = 1$. Moreover, the fact that $x_3 = s_3$ and $x_4 = s_4$ for any point x on the segment implies that $x_1 + x_2 = 1 - (s_3 + s_4)$. As a result, the segment consists of all points x in the simplex such that $x_3 = s_3$, $x_4 = s_4$, $s_1 \leq x_1 \leq s_1 + d$, and $x_2 = 1 - (s_3 + s_4) - x_1$. If we substitute the lower and upper bounds for x_1 into this expression for x_2, we obtain $s_2 \leq x_2 \leq s_2 + d$.

3.3.1 Pair-isolating side-parallel cuts

The first cutting scheme that we study, which we denote by P_s, uses only pair-isolating side-parallel cuts. Each of the three slices comprising the pair-side cut used to partition the simplex is chosen by selecting a slice distance uniformly at random from $[0, 1]$. For this cutting scheme, we prove the following lemma.

Lemma 3.1. *The maximum density of P_s is $4/3$.*

For each of the six possible basis vectors of the pair-isolating slice, we compute the expected number of times the segment is cut by the pair-side cut, conditional on the event that the vector is chosen as the basis vector of the pair-isolating slice.

Case 1: Basis vector n_{12}

The pair-isolating slice is $\Delta_{x_1+x_2=\rho_p}$, where ρ_p is chosen uniformly at random from $[0, 1]$. This slice, which isolates the pair of vertices 1, 2 from the pair 3, 4, is a hyperplane parallel to the edge that connects vertices 1 and 2. Since a $1, 2$-aligned segment is also parallel to the edge that connects vertices 1 and 2, the slice hyperplane is parallel to the segment and therefore cannot cut it. An alternative way to see this is to note that $x_1 + x_2 = 1 - (s_3 + s_4)$ for any point x on the segment. Because s_3 and s_4 are constants, the quantity $x_1 + x_2$ is fixed for any point on the segment. This implies that the pair-isolating slice cannot cut the segment, because different points on the segment can lie in the different regions $\Delta_{x_1+x_2<\rho_p}$ and $\Delta_{x_1+x_2\geq\rho_p}$ produced by the pair-isolating slice only if $x_1 + x_2 < \rho_p$ for some points x and $x_1 + x_2 \geq \rho_p$ for others. The segment will lie entirely in one of the two regions on either side of the pair-isolating slice.

Because the entire segment is in one of the regions resulting from the pair-isolating slice, it can only be cut by a side-parallel slice used to divide the region in which it lies. If it is in the source region $\Delta_{x_1+x_2<\rho_p}$, then the only slices that can cut it are side-parallel slices for a reference vertex 3 or 4, as vertices 3 and 4 are the two vertices in the source region. A $1,2$-aligned segment, however, can only be cut by side-parallel slices with a reference vertex of 1 or 2, and thus the segment cannot be cut by the 4-way cut of the simplex when the basis vector of the pair-isolating slice is n_{12} and the segment is in the source region.

If the segment is in the sink region $\Delta_{x_1+x_2\geq\rho_p}$, however, then the reference vertex for the side-parallel slices in its region will be either 1 or 2. Regardless of which of these two vertices is selected as the reference vertex, the slice may cut the segment. If the reference vertex is 1, then the segment will be cut by the side-parallel slice if the slice distance ρ_t is chosen such that $s_1 \leq \rho_t \leq s_1 + d$, as all points x on the segment satisfy $s_1 \leq x_1 \leq s_1 + d$. Similarly, if the reference vertex is 2, then the segment is cut if $s_2 \leq \rho_t \leq s_2 + d$. Figure 3-4 illustrates the ranges of slice distances for which the side-parallel slice will cut the segment. Since the slice distance is chosen uniformly at random from $[0, 1]$, in either case the probability that the segment is cut is $d/1 = d$. Thus, the expected number of times the segment is cut when the basis vector of the pair-isolating slice is n_{12} and the segment falls in the sink region is d.

We use the fact that the slice distance for the pair-isolating slice is chosen uniformly at random from $[0, 1]$ to determine the probability that the segment is in the sink region produced by the pair-isolating slice. For any point x on the segment, $x_1 + x_2 = 1 - (s_3 + s_4)$. As a result, the segment is in the sink region if the slice distance ρ_p for the pair-isolating slice is less than or equal to $x_1 + x_2$. Figure 3-5 shows the location of the segment in relation to pair-isolating slices of various slice distances. Because $\rho_p \leq x_1 + x_2$ if and only if $\rho_p \leq 1 - (s_3 + s_4)$ and ρ_p is distributed uniformly at random over $[0, 1]$, the probability that the segment falls in the sink region created by the pair-isolating slice is $(1 - (s_3 + s_4))/1 = 1 - (s_3 + s_4)$. We conclude that, conditional on the choice of n_{12} as the basis vector of the pair-isolating

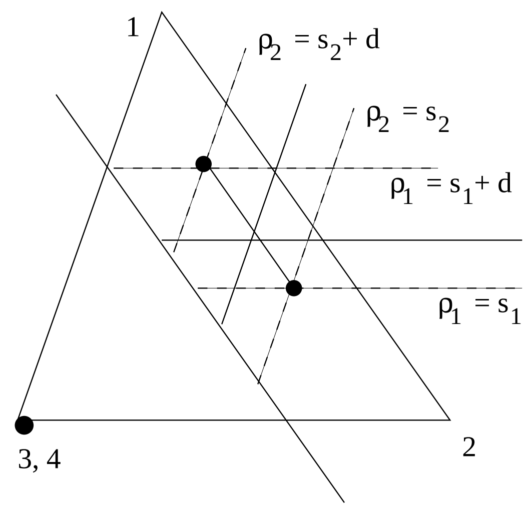


Figure 3-4: If the basis vector of the pair-isolating slice is n_{12} and the segment is in the sink region, the side-parallel slice in the sink region cuts the segment if the reference vertex is 1 and $s_1 \leq \rho_t \leq s_1 + d$, or if the reference vertex is 2 and $s_2 \leq \rho_t \leq s_2 + d$.

slice, the expected number of times the segment is cut is $(1 - (s_3 + s_4))d$.

Case 2: Basis vector n_{34}

The case in which the basis vector of the pair-isolating slice is n_{34} is symmetric to the n_{12} case, with the roles of the source and sink regions reversed in the two cases. As a result, the expected number of times the segment is cut, conditional on the choice of n_{34} as the basis vector of the pair-isolating slice, is again $(1 - (s_3 + s_4))d$.

Case 3: Basis vector n_{13}

In this case, the pair-isolating slice is $\Delta_{x_1+x_3=\rho_p}$ for a slice distance ρ_p distributed uniformly at random over $[0, 1]$. For any point x on the segment, $x_3 = s_3$ and $s_1 \leq x_1 \leq s_1 + d$, and so $s_1 + s_3 \leq x_1 + x_3 \leq s_1 + s_3 + d$. Because the segment contains all the points x in the simplex with $x_3 = s_3$, $x_4 = s_4$, and $s_1 \leq x_1 \leq s_1 + d$, it will be cut by the pair-isolating slice if and only if $s_1 + s_3 \leq \rho_p \leq s_1 + s_3 + d$. If $s_1 + s_3 + d < \rho_p$, then the segment lies entirely in the source region, while if $\rho_p < s_1 + s_3$, the segment lies entirely in the sink region. Since the slice distance ρ_p is chosen uniformly at random from $[0, 1]$, the segment is cut with probability $d/1 = d$, falls in the source region with probability $(1 - (s_1 + s_3 + d))/1 = 1 - (s_1 + s_3 + d)$, and falls in the sink

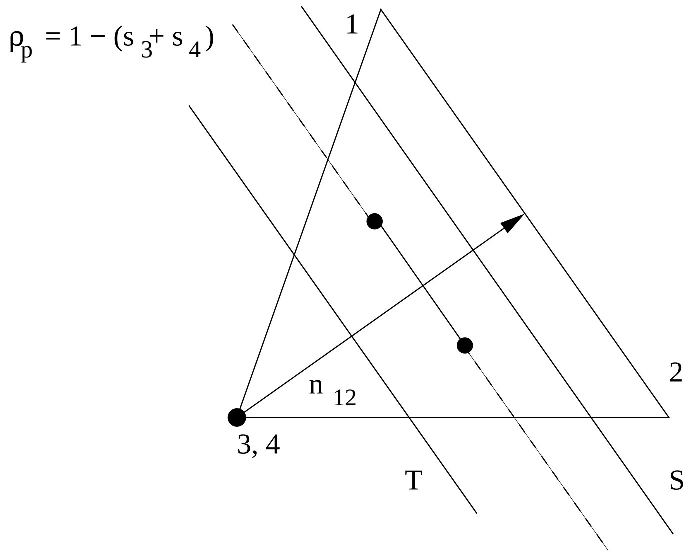


Figure 3-5: The basis vector of the pair-isolating slice is n_{12}. If the pair-isolating slice is a hyperplane such as S with a slice distance ρ_p greater than $1 - (s_3 + s_4)$, then the segment lies in the source region. On the other hand, if the pair-isolating slice is a hyperplane such as T that has a slice distance less than $1 - (s_3 + s_4)$, then the segment is in the sink region.

region with probability $(s_1 + s_3)/1 = s_1 + s_3$.

Suppose that the entire segment ends up in the source region. The source region contains vertices 2 and 4, so the segment can be cut only if the reference vertex for the side-parallel slice in the source region is vertex 2. Because each of the two candidates for the reference vertex is chosen with probability 1/2, the probability that vertex 2 is selected as the reference vertex is 1/2. If vertex 2 is the reference vertex, then the segment is cut if the slice distance ρ_s for the side-parallel slice in the source region is in the range $s_2 \leq \rho_s \leq s_2 + d$. The probability of this occurrence is $d/1 = d$. Therefore, if the basis vector of the pair-isolating slice is n_{13} and the segment falls in the source region, the expected number of times it is cut is $(1/2)d = d/2$. The case in which the segment is in the sink region is entirely analogous. In this case, the segment can only be cut if the reference vertex is 1 and the slice distance ρ_t satisfies $s_1 \leq \rho_t \leq s_1 + d$. Again, the expected number of times the segment is cut is $d/2$.

If the segment is cut by the pair-isolating slice, then it is divided such that subsets of it lie in both regions, and so it may be cut by side-parallel slices in both regions. Let γ be the quantity that satisfies $\rho_p = s_1 + s_3 + \gamma d$. We may consider γ to be the fraction of the segment that is in the source region when the simplex is partitioned according to the pair-isolating slice. Since $x_3 = s_3$ for all points x on the segment, the point at which the pair-isolating slice intersects the segment is the point x on the segment with $x_1 = s_1 + \gamma d$. By the fact that $x_2 = 1 - (s_3 + s_4) - x_1$ for points on the segment, this point has $x_2 = 1 - s_1 - s_3 - s_4 - \gamma d = s_2 + d - \gamma d = s_2 + d(1 - \gamma)$. Points x on the segment with $x_1 < s_1 + \gamma d$ and $x_2 > s_2 + d(1 - \gamma)$ are in the source region, while points x on the segment with $x_1 \geq s_1 + \gamma d$ and $x_2 \leq s_2 + d(1 - \gamma)$ are in the sink region. Figure 3-6 shows the boundaries of the portions of the segment in each of the regions.

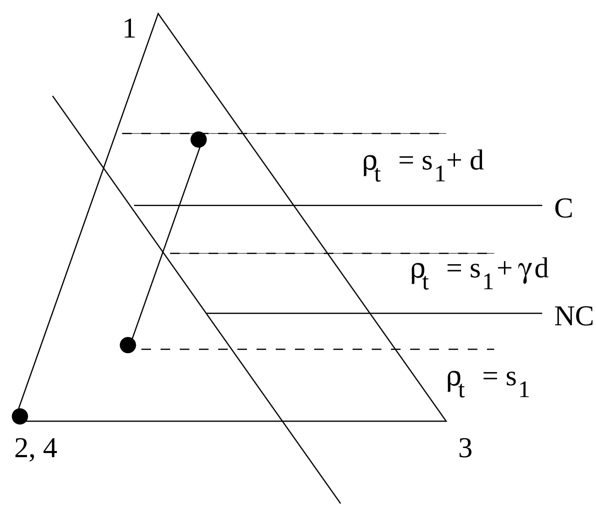


Figure 3-6: The basis vector of the pair-isolating slice is n_{13}. All the points x in the sink region satisfy $x_1 \geq s_1 + \gamma d$. As a result, a side-parallel slice in the sink region with a slice distance ρ_t in the range $s_1 \leq \rho_t < s_1 + \gamma d$, such as NC, does not cut the segment. In order for the slice in the sink region to cut the segment, it must satisfy $s_1 + \gamma d \leq \rho_t \leq s_1 + d$, as does C.

In the source region, the segment may be cut if the reference vertex is vertex 2 and the slice distance ρ_s is in the range $s_2 \leq \rho_s \leq s_2 + d$. Because all of the points on the segment that lie in the source region have $x_2 > s_2 + d(1 - \gamma)$, however, the slice distance must satisfy $s_2 + d(1 - \gamma) < \rho_s \leq s_2 + d$ in order for the side-parallel slice to cut the segment. The points on the segments with coordinate values x_2 less than $s_2 + d(1 - \gamma)$ have been captured by the pair-isolating slice, ending up in the sink region, and therefore cannot be cut by a side-parallel slice in the source region. As the reference vertex is vertex 2 with probability $1/2$ and ρ_s is chosen uniformly at random from $[0, 1]$, we conclude that the expected number of times the segment is cut by the side-parallel slice in the source region is $(1/2)((s_2 + d) - (s_2 + d(1 - \gamma)))/1 = (\gamma d)/2$.

For the sink region, the calculation of the expected number of times the segment is cut by the side-parallel slice is similar. The reference vertex must be vertex 1 for the segment to be cut, and all points x in the region satisfy $x_1 \geq s_1 + \gamma d$. Therefore, the segment is cut only if the slice distance ρ_t is in the range $s_1 + \gamma d \leq \rho_t \leq s_1 + d$. Figure 3-6 illustrates the ranges of slice distances for which the side-parallel slice cuts the segment. The expected number of times the segment is cut by the side-parallel slice in the sink region is $(1/2)((s_1 + d) - (s_1 + \gamma d))/1 = (d/2)(1 - \gamma)$. By linearity of expectation, the expected number of times the segment is cut by side-parallel slices is $(\gamma d)/2 + (d/2)(1 - \gamma) = d/2$. As the cut by the pair-isolating slice itself increases this count of the number of cuts by one, when the segment is cut by the pair-isolating slice, the expected number of times the segment is cut by the 4-way cut of the simplex is $1 + d/2$.

Now that we have the probability that the segment is cut by the pair-isolating slice, the probabilities that it lies in each of the regions produced by that slice, and the expected number of times it is cut in all of these cases, we use the total probability theorem to compute the expected number of times the segment is cut, conditional on the choice of n_{13} as the basis vector of the pair-isolating slice. This quantity is $(d)(1+d/2)+(1-(s_1+s_3+d))(d/2)+(s_1+s_3)(d/2) = d(1+d/2)+(d/2)(1-d) = (3/2)d$.

Cases 4, 5, 6: Basis vectors n_{14}, n_{23}, and n_{24}

The cases in which the basis vector of the pair-isolating slice is n_{14}, n_{23}, or n_{24} are all symmetric to the case of n_{13}. In each of these instances, one of the vertices in each region produced by the pair-isolating slice is either vertex 1 or vertex 2, and the remaining vertex in the region is either vertex 3 or vertex 4. For a $1, 2$-aligned segment, there is a symmetry between coordinates 1 and 2 and a symmetry between coordinates 3 and 4 that together cause the calculation of the expected number of times the segment is cut for the basis vector n_{14}, n_{23}, or n_{24} to be symmetric to the calculation for the basis vector n_{13}. In all four cases, the expected number of times the segment is cut is $(3/2)d$.

Conditioning on choice of basis vector

Because each of the six possible basis vectors is chosen as the basis vector of the pair-isolating slice with probability $1/6$, by the total probability theorem the expected number of times the segment is cut by a 4-way cut of the simplex chosen according to the cutting scheme is $(1/6)(2(1-(s_3+s_4))d+4(3/2)d) = (1/6)d(8-2(s_3+s_4)) = d(4/3-(1/3)(s_3+s_4))$. Therefore, the density of the cutting scheme on a segment s of length d in the simplex with endpoints (s_1+d, s_2, s_3, s_4) and (s_1, s_2+d, s_3, s_4) is $4/3-(1/3)(s_3+s_4)$. Since $s_3+s_4 \geq 0$ for all points in the simplex, and there are points in the simplex with $s_3+s_4=0$, the maximum density of this cutting scheme, in which the slice distances for the pair-isolating slice and the side-parallel slices are all distributed uniformly at random over $[0,1]$, is $4/3$.

3.3.2 Pair-isolating edge-perpendicular cuts

The second cutting scheme that we analyze is a probability distribution P_e over pair-isolating edge-perpendicular cuts. Again, the slice distance for each slice is chosen uniformly at random from the range of possible values. While the slice distance ρ_p for the pair-isolating slice, as in the pair-side cutting scheme, is drawn from $[0,1]$, the slice distances ρ_s and ρ_t for the edge-perpendicular slices are both drawn from $[-1,1]$. We calculate the maximum density of the cutting scheme, obtaining the following result.

Lemma 3.2. *The maximum density of P_e is $4/3$.*

Our analysis is very similar to the analysis for the pair-side cutting scheme. We again separate the analysis into cases based on the choice of the basis vector of the pair-isolating slice, and compute the expected number of times the segment s is cut in each case. Because the pair-isolating slice is chosen from the same distribution in these two cutting schemes, all of our analysis of the pair-isolating slice for the pair-side cutting scheme holds for the pair-edge cutting scheme as well. In particular, the probability that the segment is cut by the pair-isolating slice and the probabilities that it falls in each of the two regions produced by the pair-isolating slice are all equal to the corresponding probabilities for the pair-side cutting scheme. Thus, it remains to compute the expected number of times the segment is cut by the edge-perpendicular slices for the various cases.

First, consider the case that the basis vector of the pair-isolating slice is n_{12}. The segment cannot be cut by the pair-isolating slice, so it falls entirely in one of the two regions produced by the pair-isolating slice. If the segment is in the source region, then the only edge-perpendicular slice that may cut it is the one for the source region. The edge connecting any pair of vertices of the 4-simplex is perpendicular to the edge connecting the other pair of vertices. For this reason, a $1,2$-aligned segment is perpendicular to the edge joining vertices 3 and 4. Since the source region contains vertices 3 and 4, and an edge-perpendicular slice in a region containing vertices 3 and 4 is a hyperplane normal to the edge joining vertices 3 and 4, it follows that the edge-perpendicular slice hyperplane in the source region is parallel to a $1,2$-aligned segment and thus cannot cut it. An alternative way of reaching this conclusion is to note that the edge-perpendicular slice in the source region will be $\Delta_{x_3-x_4=\rho_s}$ or $\Delta_{x_4-x_3=\rho_s}$ for some slice distance ρ_s. Because $x_3 = s_3$ and $x_4 = s_4$ for all points x on the segment, the two quantities $x_3 - x_4 = s_3 - s_4$ and $x_4 - x_3 = s_4 - s_3$ are both fixed for all points on the segment. Therefore, the segment cannot be cut by an edge-perpendicular slice in a region that contains vertices 3 and 4, and as a result the segment lies entirely in one of the two partitions induced by the edge-perpendicular slice.

The edge-perpendicular slice in the sink region is $\Delta_{x_1-x_2=\rho_t}$ or $\Delta_{x_2-x_1=\rho_t}$ for some slice distance ρ_t. This slice may cut the segment if the segment is in the sink region. Because any point x on the segment satisfies $s_1 \le x_1 \le s_1+d$ and $x_2 = 1-(s_3+s_4)-x_1$, the quantities x_1-x_2 and x_2-x_1 are in the ranges $s_1-s_2-d \le x_1-x_2 \le s_1-s_2+d$ and $s_2-s_1-d \le x_2-x_1 \le s_2-s_1+d$. Furthermore, since the segment consists of all points x in the simplex with $x_3 = s_3$, $x_4 = s_4$, $s_1 \le x_1 \le s_1+d$, and $x_2 = 1-(s_3+s_4)-x_1$, it will be cut if and only if the basis vector is $e^1 - e^2$ and $s_1 - s_2 - d \leq \rho_t \leq s_1 - s_2 + d$, or the basis vector is $e^2 - e^1$ and $s_2 - s_1 - d \leq \rho_t \leq s_2 - s_1 + d$. Figure 3-7 demonstrates the hyperplanes that cut the segment for the basis vector $e^2 - e^1$. We use the fact that the slice distance ρ_t for the edge-perpendicular slice is distributed uniformly at random over $[-1, 1]$ to conclude that the expected number of times the segment is cut, regardless of the choice of the basis vector, is $(2d)/2 = d$. The probability that the segment falls in the sink region produced by the pair-isolating slice is $1 - (s_3 + s_4)$, and so the expected number of times the segment is cut, conditional on the choice of n_{12} as the basis vector of the pair-isolating slice, is $(1 - (s_3 + s_4))d$. An analogous calculation shows that, as in the pair-side cutting scheme, the expected number of times the segment is cut when the basis vector of the pair-isolating slice is n_{34} is also $(1 - (s_3 + s_4))d$.

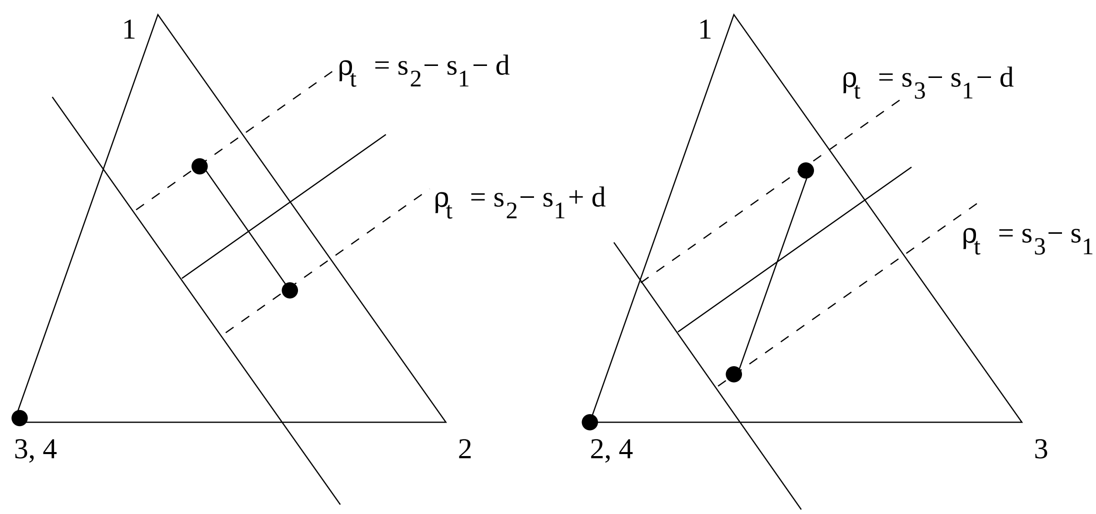


Figure 3-7: The ranges of slice distances for which edge-perpendicular slices cut segments. On the left, the basis vector of the edge-perpendicular slice is $e^2 - e^1$, and the hyperplane cuts the segment if its slice distance is in the range $s_2 - s_1 - d \leq \rho_t \leq s_2 - s_1 + d$. The right diagram shows the separation of vertices 1 and 3 in the sink region. The basis vector of the edge-perpendicular slice is $e^3 - e^1$, and the slice cuts the segment if $s_3 - s_1 - d \leq \rho_t \leq s_3 - s_1$.

If the basis vector of the pair-isolating slice is n_{13}, then the segment may be cut by the pair-isolating slice or may lie entirely in one of the two regions induced by the slice. Suppose that the segment is in the source region. The source region contains vertices 2 and 4, and so the edge-perpendicular slice is either $\Delta_{x_2-x_4=\rho_s}$ or $\Delta_{x_4-x_2=\rho_s}$. For any point x on the segment, $s_2 \leq x_2 \leq s_2 + d$ and $x_4 = s_4$, and so the segment is cut if and only if the basis vector of the edge-perpendicular slice is $e^2 - e^4$ and $s_2 - s_4 \leq \rho_s \leq s_2 - s_4 + d$, or the basis vector is $e^4 - e^2$ and $s_4 - s_2 - d \leq \rho_s \leq s_4 - s_2$. In either of these two cases, the range of values of the slice distance ρ_s that will lead to a cut of the segment by the edge-perpendicular slice is d, and so the probability that the segment is cut by the edge-perpendicular slice is $d/2$. A similar calculation shows that if the segment is in the sink region, then it is cut by the edge-perpendicular slice in that region with probability $d/2$. Figure 3-7 includes an illustration of this situation.

In the case that the segment is cut by the pair-isolating slice, we again set γ such that $\rho_p = s_1 + s_3 + \gamma d$. For points x on the segment in the source region, $x_1 < s_1 + \gamma d$ and $x_2 > s_2 + d(1-\gamma)$, and thus the segment is cut by the edge-perpendicular slice in the source region if and only if the basis vector of the edge-perpendicular slice is $e^2 - e^4$ and $s_2 - s_4 + d(1-\gamma) < \rho_s \leq s_2 - s_4 + d$, or the basis vector is $e^4 - e^2$ and $s_4 - s_2 - d \leq \rho_s < s_4 - s_2 - d(1-\gamma)$. We conclude that the expected number of times the edge-perpendicular slice in the source region cuts the segment is $(d - d(1-\gamma))/2 = (\gamma d)/2$. Similarly, points x in the sink region satisfy $x_1 \geq s_1 + \gamma d$ and $x_2 \leq s_2 + d(1-\gamma)$, and the segment is cut by the edge-perpendicular slice in the sink region if and only if the basis vector of the slice is $e^1 - e^3$ and $s_1 - s_3 + \gamma d \leq \rho_t \leq s_1 - s_3 + d$, or the basis vector is $e^3 - e^1$ and $s_3 - s_1 - d \leq \rho_t \leq s_3 - s_1 - \gamma d$. The expected number of times that the edge-perpendicular slice in the sink region cuts the segment is therefore $(d - \gamma d)/2 = (d/2)(1-\gamma)$. As a result, the expected number of times the segment is cut by the 4-way cut when the pair-isolating slice cuts it is $1 + (\gamma d)/2 + (d/2)(1-\gamma) = 1 + d/2$.

We now use the total probability theorem to compute the expected number of times the segment is cut when the basis vector of the pair-isolating slice is n_{13} as $d(1+d/2) + (1-(s_1+s_3+d))(d/2) + (s_1+s_3)(d/2) = d(1+d/2) + (1-d)(d/2) = (3/2)d$. Again, the cases in which the basis vector of the pair-isolating slice is n_{14}, n_{23}, or n_{24}

are all symmetric to the case of n_{13}, and so the expected number of cuts of the segment in each case is $(3/2)d$.

As in the pair-side cutting scheme, we compute the density of the pair-edge cutting scheme on the segment s by applying the total probability theorem, using the fact that each of the possible basis vectors of the pair-isolating slice is chosen with probability 1/6. Because the expected number of times the segment is cut for each of the basis vectors is the same for the pair-edge cutting scheme as for the pair-side cutting scheme, the calculation of the density for the pair-edge cutting scheme is identical to that of the pair-side cutting scheme and the result is a density of $4/3 - (1/3)(s_3 + s_4)$. This quantity leads to the conclusion that the maximum density of this pair-isolating edge-perpendicular cutting scheme in which the slice distances are all distributed uniformly at random over the ranges of possible values is 4/3.

3.3.3 Comparison to sparc cutting scheme with uniform slice distances

The analogue for sparcs of the pair-isolating cutting schemes that we have analyzed is a cutting scheme in which all of the slice distances ρ_i for the side-parallel slices are drawn uniformly at random from $[0, 1]$. One may show that this cutting scheme has a maximum density of 5/4, which is smaller than the 4/3 of these pair-isolating cutting schemes with uniform slice distances. As such, when slice distances are all chosen uniformly at random from the ranges of possible values, sparcs yield a better approximation ratio for the Călinescu et al. algorithm than pair-side cuts and pair-edge cuts. This fact could be interpreted as a suggestion that for cutting schemes that are distributions over a single type of cut, sparcs are more effective than pair-isolating cuts.

It is worth noting that the density of the pair-isolating cutting schemes on segments is not uniform over the simplex. The cutting scheme over sparcs in which the slice distances are all selected uniformly at random from $[0, 1]$ also has this property of non-uniform density over the simplex. In the case of pair-isolating cutting schemes, the density of each scheme we have studied on a $1, 2$-aligned segment (x, y) whose endpoints have coordinate values $x_3 = y_3 = s_3$ and $x_4 = y_4 = s_4$ is $4/3-(1/3)(s_3+s_4)$. Thus, the density is maximized for segments with $s_3+s_4 = 0$, and it decreases linearly as $s_3 + s_4$ increases.

Our analysis of the pair-isolating slice in the cutting schemes implies that for a point x, the quantity $x_3 + x_4$ may be considered to be the distance of the point x, measured along a vector that goes from the midpoint of the edge that connects vertices 1 and 2 to the midpoint of the edge that connects vertices 3 and 4. As a result, the smaller the value of $s_3 + s_4$, the closer the segment is to the edge that joins vertices 1 and 2. We conclude that the density of the pair-side and pair-edge cutting schemes on a $1, 2$-aligned segment is a function of the distance of the segment from the edge that connects vertices 1 and 2. The closer the segment is to the edge joining vertices 1 and 2, the higher the density of the cutting schemes on the segment. Figure 3-8 shows the relationship between the distance from the and the edge connecting vertices 1 and 2 to the segment and the density of the cutting schemes on the segment. By the symmetry of the cutting scheme, an analogous relationship holds for any i, j-aligned segment with the edge connecting vertices i and j.

For an i, j-aligned segment, the closer the segment is to the edge that connects vertices i and j, the greater the density of the cutting schemes on the segment. In particular, the density of the cutting schemes on the edge joining vertices i and j is the maximum density of the cutting schemes. For the sparc cutting scheme in which the slice distances are chosen uniformly at random from $[0, 1]$, as well as for the cutting schemes using pair-isolating cuts, the maximum density of the scheme on the edges joining the simplex vertices is the maximum density of the cutting scheme.

Consider a cutting scheme in which a sparc is chosen with probability p, and a pair-isolating cut is chosen with probability $1 - p$. Regardless of which type of cut is chosen, all the slice distances are chosen uniformly at random from the range of possible values. For any segment in the simplex, the density of this cutting scheme on the segment is a convex combination of the density of the sparc cutting scheme on it and the density of the cutting scheme involving pair-isolating cuts on it. If the

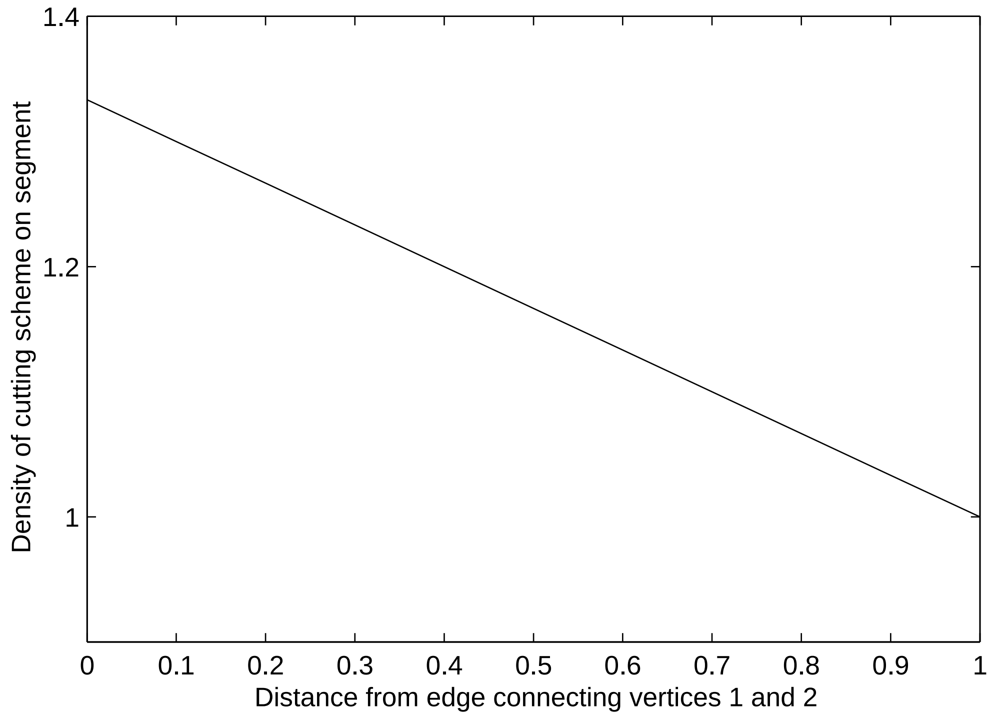


Figure 3-8: Density of the two cutting schemes using uniform pair-isolating cuts on a $1, 2$-aligned segment as a function of the distance from the edge connecting vertices 1 and 2 to the segment.

sparc cutting scheme had high density on segments on which the cutting scheme using pair-isolating cuts had low density, and the cutting scheme involving pair-isolating cuts had high density on segments on which the sparc cutting scheme had low density, then the combined cutting scheme could have a maximum density lower than both the maximum densities of the individual cutting schemes. In general, however, the density of the sparc cutting scheme on segments in the simplex varies over the simplex in much the same way as the density of the cutting schemes involving pair-isolating cuts. Furthermore, because the densities of both individual cutting schemes on the edges connecting vertices of the simplex are the maximum densities of the respective schemes, any cutting scheme that chooses a sparc with probability p and a pair-isolating cut with probability $1-p$ and selects all slice distances uniformly at random from the range of possible values must have a maximum density at least as large as the smaller maximum density of the two individual cutting schemes, $5/4$.

Chapter 4

Computational Experiments Involving Pair-Isolating Cuts

The focus of our work is an extension of the computational experiments of Karger et al. to use pair-isolating 4-way cuts of the 4-simplex in addition to side-parallel cuts. In this chapter we describe how we introduce pair-isolating side-parallel and pair-isolating edge-perpendicular cuts into the linear programs developed by Karger et al. to study cutting schemes. We present the results of our experiments and some observations about the solutions to the discrete linear programs.

Our experimental results suggest that for cutting schemes involving a single type of cut, sparcs are preferable to pair-isolating cuts because they yield smaller approximation ratios in the Călinescu et al. algorithm. On the other hand, cutting schemes including both sparcs and pair-isolating cuts have smaller maximum densities than schemes that use a single type of cut, and therefore pair-isolating cuts are apparently useful in combination with sparcs for improving the cutting schemes obtained from the solutions to the discrete linear programs. In the results of our experiments, cutting schemes involving pair-isolating side-parallel cuts generally lead to smaller approximation ratios than schemes involving pair-isolating edge-perpendicular cuts.

4.1 Adding Pair-Isolating Cuts to the Linear Program

Karger et al. developed a discrete linear program that generates cutting schemes that yield small approximation ratios when used in the Călinescu algorithm, and provides computational proofs of upper bounds on the maximum densities of these cutting schemes. To ensure that the objective value of the primal linear program was at least as large as the maximum density of the cutting scheme specified by a solution to the program, they computed the constraints using upper bounds on the density induced by specific cuts on segments. When we introduce pair-isolating cuts into the linear program, our principal task is to compute an upper bound on the density of a particular pair-isolating cut on any segment in a particular cell of the simplex, so that the objective value of the linear program for a solution will remain an upper bound on the maximum density of the cutting scheme corresponding to the solution.

4.1.1 Discrete pair-isolating cuts

To limit the number of pair-isolating cuts represented by variables in the linear program, we define discrete pair-isolating cuts in much the same way as Karger et al. define discrete sparcs. We add variables to the linear program that correspond to discrete pair-isolating cuts, with the value of a variable specifying the probability of choosing the associated pair-isolating cut. As in our analysis of the pair-side and pair-edge cutting schemes with uniform distributions over the slice distances, we only consider pair-isolating cuts in which the basis vector of the pair-isolating slice is chosen uniformly at random from the set of six possible basis vectors, the reference vertex of each side-parallel slice is chosen uniformly at random from the set of two candidates, and the basis vector of each edge-perpendicular slice is chosen uniformly at random from the set of two possibilities. Because each pair-isolating cut we consider chooses basis vectors and reference vertices from the same distribution, individual cuts differ only in the slice distances ρ_p, ρ_s, and ρ_t of the slices that partition the simplex.

A *discrete pair-isolating cut* is a triple (u_p, u_s, u_t) of integers. For simplicity and ease of calculation, we use a common grid size N to partition the ranges over which the slices are drawn for both pair-isolating slices and side-parallel slices. On the other hand, we found that, in the interests of obtaining small objective values from the solutions to the primal linear programs, it was necessary to divide the range $[-1, 1]$ from which edge-perpendicular slices are drawn into $2N$ slabs, giving each slab a length of $1/N$. Figure 4-1 illustrates the slabs for pair-isolating and edge-perpendicular slices. The integers u_p, u_s, and u_t for a discrete pair-isolating side-parallel cut are all in the range $[0, N-1]$. For a discrete pair-isolating edge-perpendicular cut, u_p is an integer in the range $[0, N-1]$, and u_s and u_t are integers in $[0, 2N-1]$.

From a discrete pair-isolating cut, we obtain a distribution over continuous pair-isolating cuts by applying the conventions established by Karger et al. for sparcs. We choose the slice distance ρ_p for the pair-isolating slice by choosing ρ_p uniformly at random from $[u_p/N, (u_p+1)/N]$. For pair-isolating side-parallel cuts, we set the slice distances ρ_s and ρ_t for the side-parallel slices in the source and sink regions, respectively, by choosing ρ_s uniformly at random from $[u_s/N, (u_s+1)/N]$ and selecting ρ_t uniformly at random from $[u_t/N, (u_t+1)/N]$. In the case of pair-isolating edge-perpendicular slices, ρ_s is chosen uniformly at random from $[-1+u_s/N, -1+(u_s+1)/N]$ and ρ_t is chosen uniformly at random from $[-1+u_t/N, -1+(u_t+1)/N]$.

Discrete pair-isolating cuts exhibit symmetry that we exploit to reduce the number of discrete cuts that we must include in the linear programs. Suppose the vertices of the simplex are paired such that one pair contains vertices v_1 and v_2, and the other pair has vertices w_1 and w_2. Because the slice distances are chosen uniformly at random from the slab in which they fall, when the basis vector of the pair-isolating slice is $n_{v_1 v_2}$ and the pair-isolating slice slab is $u_p = i$, the slice distance of the pair-isolating slice has the same conditional distribution as it would have if the basis vector were $n_{w_1 w_2}$ and the slice slab were $(N-1)-i$. Figure 4-1 illustrates this situation. As a result of this symmetry, the source slab in the case of the basis vector $n_{v_1 v_2}$ will be determined by the same distribution as the sink slab in the case of the basis vector $n_{w_1 w_2}$, and the same relationship holds for the sink region for $n_{v_1 v_2}$ and the source region for $n_{w_1 w_2}$. Since each of the candidate basis vectors is chosen as the basis vector of the pair-isolating slice with equal probability, it follows that for any discrete pair-isolating cut (u_p, u_s, u_t), there is a corresponding discrete cut $((N-1)-u_p, u_t, u_s)$ that will have the same density as (u_p, u_s, u_t) on any segment in the simplex. For this reason, we need only include discrete pair-isolating cuts with $u_p < N/2$ in the discrete linear program.

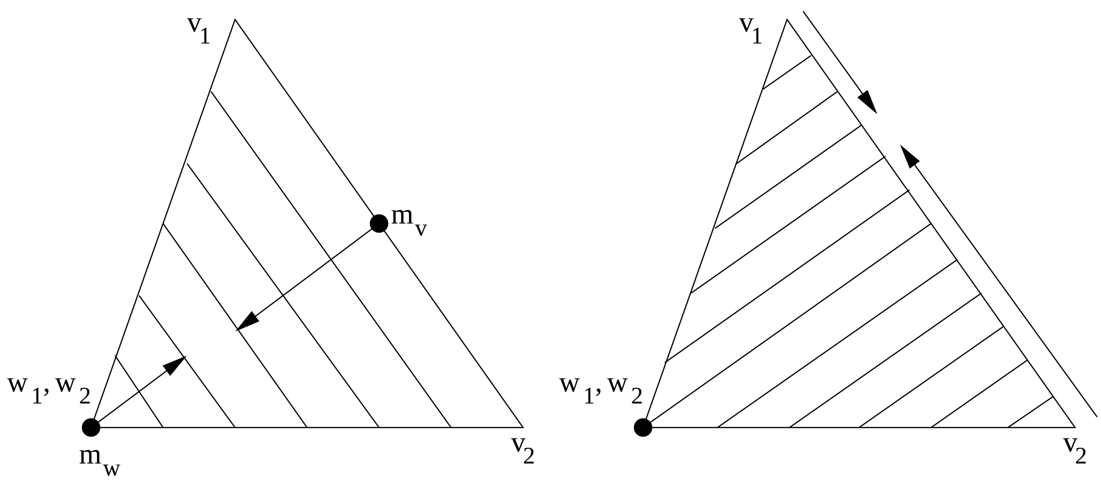


Figure 4-1: Symmetries of discrete pair-isolating cuts. In the left diagram, the grid size is $N = 6$, and the slab $u_p = 2$ for the basis vector $n_{v_1 v_2}$ of the pair-isolating slice is the same as the slab $u_p = (N-1) - 2 = 3$ for the basis vector $n_{w_1 w_2}$. On the right, there are $2N = 12$ slabs, and the slab $u_i = 3$ for the basis vector $e^{v_2} - e^{v_1}$ of an edge-perpendicular slice is the same as the slab $u_i = (2N-1) - 3 = 8$ for the basis vector $e^{v_1} - e^{v_2}$.

Edge-perpendicular slices also have a symmetry that allows us to reduce the number of discrete cuts that we consider. For an edge-perpendicular slice with a slice distance ρ_i that separates vertices v_1 and v_2, either $e^{v_2} - e^{v_1}$ or $e^{v_1} - e^{v_2}$ is the basis vector. If the basis vector is $e^{v_2} - e^{v_1}$ and ρ_i is drawn from the slab specified by $u_i = i$, then the slice distance of the edge-perpendicular slice has the same conditional distribution as it would have if the basis vector were $e^{v_1} - e^{v_2}$ and the slice were in the slab $(2N-1) - i$. Figure 4-1 shows an example of this symmetry. As each of the two vectors is chosen as the basis vector with probability 1/2, a discrete pair-edge cut (u_p, u_s, u_t) has the same density as another discrete pair-edge cut

$(u_p, (2N-1)-u_s, (2N-1)-u_p)$. Thus, we consider only discrete pair-edge cuts that have $u_s < N$ and $u_t < N$.

4.1.2 Cells of the simplex

In the discrete linear program developed by Karger et al., the side-parallel slice slabs partition the k-simplex into cells. For each of the k coordinates i, dividing the simplex into the slabs $\{x \mid j/N \leq x_i \leq (j+1)/N\}$ for $j = 0, \ldots, N-1$ produces a distinct partition of the simplex. As such, a cell is identified by k slabs q_i, and is described by $\{x \mid q_i/N \leq x_i \leq (q_i+1)/N, i = 1, \ldots, k\} = \cap_{i=1}^{k}\{x \mid q_i/N \leq x_i \leq (q_i+1)/N\}$. A cell is the intersection of k side-parallel slice slabs, one for each coordinate.

When we introduce pair-isolating cuts into the discrete linear program, we divide up the cells defined by Karger et al. for sparcs so that each cell in the linear program spans exactly one slab for each type of slice included in the program. Consider the slabs of an edge-perpendicular slice that has a basis vector of $e^{v_2} - e^{v_1}$, as depicted in Figure 4-2. For a cell that is characterized by the side-parallel slice slabs (q_1, q_2, q_3, q_4), the range of values of the quantity $x_{v_2} - x_{v_1}$ for points x in the cell determines where the cell lies in relation to the edge-perpendicular slice slabs. Since $q_{v_1}/N \leq x_{v_1} \leq (q_{v_1}+1)/N$ and $q_{v_2}/N \leq x_{v_2} \leq (q_{v_2}+1)/N$, this range of values is $(q_{v_2} - q_{v_1} - 1)/N \leq x_{v_2} - x_{v_1} \leq (q_{v_2} - q_{v_1} + 1)/N$. As an edge-perpendicular slice slab is a range of the form $[-1 + u_i/N, -1 + (u_i+1)/N]$, the cell spans the two slabs $u_i = q_{v_2} - q_{v_1} - 1 + N$ and $u_i = q_{v_2} - q_{v_1} + N$.

A cell that spans two slabs for the edge-perpendicular slice will have a weak upper bound on the density of a cutting scheme on any segment within it. To see this, consider a segment that lies entirely in one of the two pair-isolating slice slabs within the cell. This segment can only be cut by the edge-perpendicular slice when the slab u_i takes on one particular value. All cuts that have either of the two values of u_i such that the edge-perpendicular slice passes through the cell, however, cut some segments in the cell. For this reason, in order to obtain an upper bound on the density of a probability distribution over discrete pair-isolating cuts on any segment within the cell, we must assign positive upper bounds to the densities induced by cuts whose

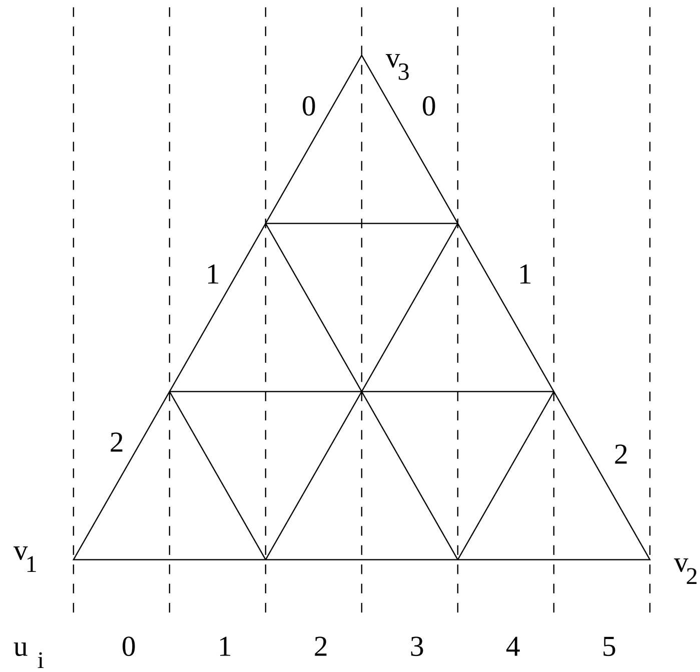


Figure 4-2: This diagram shows the face of the 4-simplex opposite vertex v_4 for the grid size $N = 3$. On the edge connecting vertices v_1 and v_3 are labels that identify the side-parallel slice slabs for coordinate 1. The edge joining vertices v_2 and v_3 has the corresponding labels for coordinate 2. Below the edge connecting vertices v_1 and v_2, the labels identify the edge-perpendicular slice slabs for the basis vector $e^{v_2} - e^{v_1}$.

edge-perpendicular slices are in either of the two slabs. The result is that the density bound computed for the cell is not tight for a segment that lies entirely in one of the slice slabs.

For a segment that contains points in both slabs for the pair-isolating slice, we may divide the segment into two subsets using the hyperplane boundary shared by the two slabs, $\Delta_{v_2 - v_1 = (q_{v_2} - q_{v_1})/N}$. We may then compute a density bound for each subset by considering only the cuts that have an edge-perpendicular slice in the one slab in which the subset falls. The density on the entire segment is at most the larger of the upper bounds for the subsets, and so we need assign positive upper bounds only to the densities of cuts whose edge-perpendicular slices are in one particular slab to obtain a valid upper bound on the total density of the cutting scheme on the segment. Thus, for all segments in the cell that spans multiple pair-isolating slice slabs, the upper bound computed for the density of a cutting scheme on any segment in the cell is not tight.

Our experiences working with the discrete linear programs showed that in order for the optimal solutions to the programs to be cutting schemes that achieve low approximation ratios, the density bounds for the cells must be tighter than the ones that arise when cells span multiple slabs for a particular type of slice. In general, for the solutions to have small maximum densities, each cell that has a constraint in the linear program must span exactly one slab for each type of slice. As a result, the cells that we account for in the discrete linear program are formed by taking the intersection of different slabs. For a particular instance of the discrete program, only the slabs for a type of slice that is included in the cuts in the program need be accounted for in the intersections of slabs, and so the cells in the constraints of the program vary from one instance to another based on the types of cuts included in the different instances.

There are six possible basis vectors of the pair-isolating slice. Note, however, that for a pair-isolating slice that isolates the pair of vertices v_1, v_2 from the pair w_1, w_2, the two possible basis vectors, $n_{v_1 v_2}$ and $n_{w_1 w_2}$, are parallel but point in opposite directions. For this reason, the slabs for the two basis vectors are the same, as illustrated in Figure 4-1. Thus, there are only three distinct sets of slabs for pair-isolating slices, one for each distinct grouping of the vertices into two pairs.

We identify a particular pairing of the vertices by specifying the vertex that is paired with vertex 1. As a result, a cell induced by pair-isolating slices is obtained by taking the intersections of the three pair-isolating slice slabs $q_{p_{12}}$, $q_{p_{13}}$, and $q_{p_{14}}$, where a slab $q_{p_{ij}}$ is the set $\{x \mid q_{p_{ij}}/N \leq x_i + x_j \leq (q_{p_{ij}} + 1)/N\}$. For the purpose of consistency, we refer to the side-parallel slice slabs that determine a cell as q_{s_1}, q_{s_2}, q_{s_3}, and q_{s_4}, where a slab q_{s_i} is $\{x \mid q_{s_i}/N \leq x_i \leq (q_{s_i} + 1)/N\}$.

In the case of edge-perpendicular slices, there are $\binom{4}{2} = 6$ distinct pairs of vertices that may be separated by these slices. As Figure 4-1 shows, for a particular pair of vertices v_1, v_2, the slabs for the basis vector $e^{v_2} - e^{v_1}$ are the same as the slabs for the basis vector $e^{v_1} - e^{v_2}$, so we need only one slab per pair of vertices to identify a cell. As a result, when computing a density bound for segments in a cell with a particular alignment, we use 6 edge-perpendicular slice slabs to determine the cell. We refer to those slabs as $q_{e_{12}}$, $q_{e_{13}}$, $q_{e_{14}}$, $q_{e_{23}}$, $q_{e_{24}}$, and $q_{e_{34}}$, where a slab $q_{e_{ij}}$ is the set $\{x \mid -1+q_{e_{ij}}/N \leq x_j - x_i \leq -1+(q_{e_{ij}}+1)/N\}$. Note that an edge-perpendicular slice that separates vertices 3 and 4 is parallel to $1,2$-aligned segments, and therefore has a density of 0 on any $1,2$-aligned segment, regardless of the range of values of $x_4 - x_3$ for points x on the segment. As such, when we compute a density bound for segments of a particular alignment in a cell, we may ignore one of the edge-perpendicular slice slabs.

To obtain a cell that has a constraint in the discrete linear program, we take the intersection of different cell slabs, including in the intersection slabs for each type of cut that the program contains. If sparcs or pair-side cuts are present in the program, we include in the intersection the side-parallel slice slabs q_{s_1}, q_{s_2}, q_{s_3}, and q_{s_4}. For a program that has pair-isolating cuts, we must include the pair-isolating slice slabs $q_{p_{12}}$, $q_{p_{13}}$, and $q_{p_{14}}$. If pair-edge cuts are present, we include in the intersection the edge-perpendicular slice slabs $q_{e_{12}}$, $q_{e_{13}}$, $q_{e_{14}}$, $q_{e_{23}}$, $q_{e_{24}}$, and $q_{e_{34}}$. A cell in a program that contains all three types of 4-way cuts of the simplex is identified by a tuple $(q_{s_1}, q_{s_2}, q_{s_3}, q_{s_4}, q_{p_{12}}, q_{p_{13}}, q_{p_{14}}, q_{e_{12}}, q_{e_{13}}, q_{e_{14}}, q_{e_{23}}, q_{e_{24}}, q_{e_{34}})$ that consists of 13 integers. The program has a constraint for each cell described by a tuple for which the intersection of the slabs identified by the integers in the tuple is not empty.

4.1.3 Upper bound on segment density

Since all of the pair-isolating cuts that we consider are symmetric with respect to all coordinates, it is sufficient to account for only $1,2$-aligned segments when we compute the upper bound on the density induced by a discrete pair-isolating cut on any segment within a cell of the simplex. Karger et al. showed how to determine an upper bound on the density of a discrete sparc on any segment within a cell by comparing the side-parallel slice slabs u_i with the cell slabs $q_{\sigma(i)}$. We perform the same type of calculation to develop an upper bound on the density of a discrete pair-isolating cut on any segment within a cell $(q_{s_1}, q_{s_2}, q_{s_3}, q_{s_4}, q_{p_{12}}, q_{p_{13}}, q_{p_{14}}, q_{e_{12}}, q_{e_{13}}, q_{e_{14}}, q_{e_{23}}, q_{e_{24}}, q_{e_{34}})$

of the 4-simplex.

Location of pair-isolating slice in relation to cell

Consider a pair-isolating slice that has a slice distance of ρ_p and a basis vector n_{ij}. Let s_1, s_2 denote the two vertices in the source region produced by the pair-isolating slice, so that i, j, s_1, s_2 is a permutation of the vertices $1, 2, 3, 4$. For a pair-isolating slice obtained from a discrete pair-isolating cut (u_p, u_s, u_t), the slice distance is in the range $\rho_p \in [u_p/N, (u_p + 1)/N]$. If the basis vector n_{ij} is one of n_{12}, n_{13}, n_{14}, then the pair-isolating slice slab for the basis vector containing the cell is $q_p = q_{p_{ij}}$. Otherwise, as depicted in Figure 4-1, the slab containing the cell is $q_p = (N-1) - q_{p_{s_1 s_2}}$. If the slab u_p containing the pair-isolating slice is not the same as the slab q_p containing the cell, then the slice does not pass through the cell. In this case, no segment in the cell can be cut by the slice, and any segment in the cell lies entirely in one of the two regions produced by the pair-isolating slice. If $u_p = q_p$, then the pair-isolating slice may go through the cell, and thus a segment in the cell may be cut by it.

Density bound for edge-perpendicular slice

In addition to the pair-isolating slice, the segments in a cell may be cut by the pair-separating slices of the pair-isolating cut. The upper bound on the density of a side-parallel slice in one of the regions of the simplex induced by the pair-isolating slice on any segment within the cell may be computed using the Karger et al. approach. For edge-perpendicular slices, we compute an upper bound on the density of a slice on any segment in the cell by again determining where the slice falls in relation to the cell.

For an edge-perpendicular slice in either the source or the sink region that is in the slab u_e and has a basis vector of $e^j - e^i$, the slice distance ρ_e is in the range $\rho_e \in [-1 + u_e/N, -1 + (u_e + 1)/N]$. If $i < j$, then the slab for the basis vector that contains the cell is $q_e = q_{e_{ij}}$. As Figure 4-1 illustrates, if $j < i$, then the cell is in the slab $q_e = q_{e_{ji}}$. Again, a segment in the cell can be cut only if the edge-perpendicular slice passes through the cell, which can occur only if $u_e = q_e$. If the slice does not pass through the cell, then the density of it on any segment in the cell is 0.

Consider a segment in the cell of length d that has endpoints (s_1+d, s_2, s_3, s_4) and (s_1, s_2+d, s_3, s_4). Let ρ_ℓ and ρ_h be the minimum and maximum values, respectively, of the quantity $x_j - x_i$ for points x on the segment. Because the slice cuts the segment if and only if $\rho_e \in [\rho_\ell, \rho_h]$, and the slice distance ρ_e is uniformly distributed over $[-1+u_e/N, -1+(u_e+1)/N]$, the probability that the slice cuts the segment is $(\rho_h - \rho_\ell)/(1/N) = N(\rho_h - \rho_\ell)$. The value of $\rho_h - \rho_\ell$ depends on the vertices i and j of the simplex. For $i = 1$ and $j = 2$, $\rho_h - \rho_\ell = 2d$. When $i \in \{1, 2\}$ and $j \in \{3, 4\}$, $\rho_h - \rho_\ell = d$. For $i = 3$ and $j = 4$, $\rho_h - \rho_\ell = 0$, and so the edge-perpendicular slice cannot cut the segment because it is parallel to it. Note that in all of these cases, swapping the values of i and j will not change the value of $\rho_h - \rho_\ell$, so these are the values of $\rho_h - \rho_\ell$ for all basis vectors $e^j - e^i$ of the edge-perpendicular slice.

Once we have the value of $\rho_h - \rho_\ell$ for the basis vector $e^j - e^i$ of the edge-perpendicular slice, we may compute the density of the slice on the segment as $(N(\rho_h - \rho_\ell))/d$. The quantity $\rho_h - \rho_\ell$ is $2d$ for $\{i, j\} = \{1, 2\}$, d for $\{i, j\} \in \{\{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}\}$, and 0 for $\{i, j\} = \{3, 4\}$. Thus, $\rho_h - \rho_\ell$ is always the product of d and a constant, and so the density has no dependence on d. It follows that $(N(\rho_h - \rho_\ell))/d$ is an upper bound on the density of the edge-perpendicular slice on any segment in the cell.

Density bound for pair-isolating cut

With this method for bounding the density of an edge-perpendicular slice on any segment in the cell, we compute an upper bound on the density of a discrete pair-isolating cut (u_p, u_s, u_t) by considering each of the six possible basis vectors of the pair-isolating slice in turn. Because each of these vectors is chosen as the basis vector with probability 1/6, the density bound will be an average of the density bounds for each of the six basis vectors. For a particular basis vector n_{ij}, we consider each of the ways of choosing the reference vertices for the side-parallel slices or the basis vectors for the edge-perpendicular slices in the two regions induced by the pair-isolating slice. As there are two possible reference vertices or basis vectors for each region, there are four cases that must be handled. In each region, each of the two possible reference vertices or basis vectors is selected with probability 1/2, and so the density bound for a particular basis vector n_{ij} of the pair-isolating slice is the average of the four density bounds for the different choices of the reference vertices for side-parallel slices or basis vectors of edge-perpendicular slices when the basis vector of the pair-isolating slice is n_{ij}.

Given a basis vector n_{ij} of the pair-isolating slice and a set of reference vertices or basis vectors for the slices that separate the two regions produced by the pair-isolating slice, we first determine the slab q_p for the pair-isolating slice that contains the cell. We then compare this slab to the slab u_p that contains the pair-isolating slice. If $q_p < u_p$, then the pair-isolating slice does not pass through the cell and the cell lies entirely in the source region produced by the pair-isolating slice. If $q_p > u_p$, then the entire cell is in the sink region. In either of these two cases, we compute a bound on the density of the slice in the region in which the cell lies on any segment in the cell using the approach of Karger et al. for a side-parallel slice, or our method for an edge-perpendicular slice. Since the density of the pair-isolating slice on any segment in the cell is 0, we use this bound as the upper bound on the density of the pair-isolating cut on any segment in the cell.

The other case to consider is $q_p = u_p$, in which the pair-isolating slice may pass through the cell. Our next step is determined by the basis vector of the pair-isolating slice. If the basis vector is n_{12} or n_{34}, as depicted in Figure 4-3, then the pair-isolating slice hyperplane is parallel to all $1, 2$-aligned segments and therefore has a density of 0 on any segment in the cell. Furthermore, a segment in the cell can be cut by a slice that separates a pair of vertices only if it is in the region containing vertices 1 and 2, which is the sink region when the basis vector of the pair-isolating slice is n_{12} and the source region when the basis vector is n_{34}. To obtain an upper bound on the density of the pair-isolating cut on any segment in the cell, we assume that the segment is in the region containing vertices 1 and 2, and we compute a bound on the density of the pair-separating slice in that region on any segment in the cell.

Suppose that $q_p = u_p$ and the basis vector of the pair-isolating slice is n_{13}. This

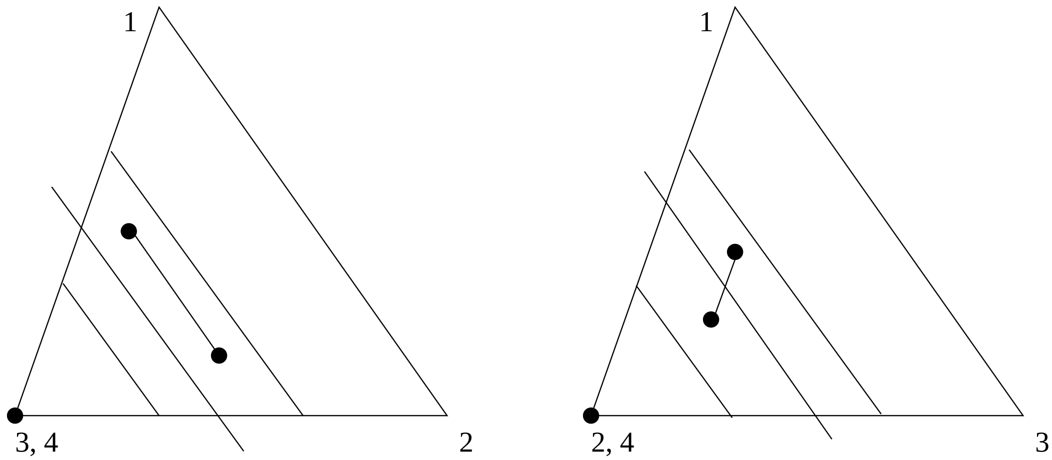


Figure 4-3: If the pair-isolating slice passes through the cell, portions of the segment may end up in both regions produced by the slice. On the left, the basis vector of the pair-isolating slice is n_{12}, and the segment can be cut only if it is in the region of the simplex containing vertices 1 and 2. In the right diagram, the basis vector of the pair-isolating slice is n_{13}. The pair-isolating slice and both pair-separating slices may cut the segment.

situation is shown in Figure 4-3. For a segment in the cell of length d with endpoints $(s_1 + d, s_2, s_3, s_4)$ and $(s_1, s_2 + d, s_3, s_4)$, the range of values of the quantity $x_1 + x_3$ for points on the segment is $x_1 + x_3 \in [s_1 + s_3, s_1 + s_3 + d]$. Because the slice distance of the pair-isolating slice is chosen uniformly at random from $[u_p/N, (u_p + 1)/N]$, the probability that the segment is cut by the pair-isolating slice is $d/(1/N) = Nd$.

Since the pair-isolating slice may pass through the cell, subsets of a segment in the cell may be in either the source or the sink region produced by the pair-isolating slice, and thus may be cut by the pair-separating slice in either of the two regions. We compute an upper bound on the density of the slice in the source region on any segment in the cell, and an upper bound on the density of the slice in the sink region on any segment in the cell. Let δ denote the maximum of the two bounds. If the pair-isolating slice does not cut a segment, then the segment lies entirely in either the source or the sink region, and so the density of the pair-isolating cut on the segment is at most δ.

In the event that the pair-isolating slice does cut a segment, the density bounds computed for the source and sink regions apply to the parts of the segment that end up in each region. Note that if any part of the segment lies in the region in which the pair-separating slice has the smaller density, then the density of the cut on the segment would increase if the pair-isolating slice were moved such that this part of the segment ended up in the region in which the slice has the larger density. As such, the density of the cut on the segment is maximized when the pair-isolating slice cuts the segment near an endpoint such that nearly all of the segment falls in the region in which the pair-separating slice has the greater density. It follows that the density of the pair-separating slices on any segment in the cell is at most δ.

By linearity of expectation, the expected number of times a segment is cut when the pair-isolating slice cuts it is at most $1 + \delta d$, as the pair-isolating slice contributes 1 cut in expectation and the pair-separating slices contribute at most δd cuts in expectation. Thus, the density of the pair-isolating cut on any segment in the cell is at most $1/d + \delta$ in the case in which the pair-isolating slice cuts the segment. Conditioning on the event that the pair-isolating slice cuts the segment, we now use the total probability theorem to compute an upper bound on the density of the pair-isolating cut on any segment in the cell of $Nd(1/d + \delta) + (1 - Nd)\delta = N + \delta$ for the case in which the basis vector is n_{13}. The cases involving the remaining basis vectors, n_{14}, n_{23}, and n_{24}, are symmetric to this case of n_{13} and therefore the computation of the upper bound on the density uses the same sequence of steps and yields the same result.

4.2 Experimental Results

To obtain the discrete primal linear programs for which the solutions are cutting schemes, we developed a program that performs the density bound computations to generate the programs automatically. We then used the optimization software CPLEX to solve the programs. The sizes of the programs that we were able to solve were limited by the finite computational resources to which we had access. For some values of the grid size N less than 30, however, we were able to solve discrete linear programs that included sparcs, pair-isolating side-parallel cuts, and pair-isolating edge-perpendicular cuts.

Table 4.1 shows the maximum densities of the cutting schemes specified by optimal solutions to the discrete primal linear programs for different combinations of types of cuts and a grid size of $N = 15$. For each combination of types of cuts, the table presents the number of cells for which the discrete program including those types of cuts has a constraint. As the discrete primal program has one variable for each cut of the simplex, a solution is a joint distribution over cuts of the simplex. For programs that contain more than one type of cut, we may capture the relative importance of the various types of cuts in the solution by computing the sum of the probabilities assigned to all the variables for each type of cut. In the table, when multiple types of cuts are included in a cutting scheme, the total probability assigned to the cuts of each type is indicated.

Cut types	Cells	Max. dens.	sparc	pair-side	pair-edge
sparc	512	1.175393	1	N/A	N/A
pair-side	1688	1.191196	N/A	1	N/A
pair-edge	17728	1.218656	N/A	N/A	1
pair-side, pair-edge	17728	1.191196	N/A	1	0
sparc, pair-side	1688	1.17233	0.729301	0.270699	N/A
sparc, pair-edge	17728	1.174643	0.881157	N/A	0.118843
sparc, pair-side, pair-edge	17728	1.172152	0.743059	0.228798	0.028143

Table 4.1: The maximum densities of the cutting schemes specified by the optimal solutions to the discrete linear programs for a grid size of $N = 15$ and various combinations of types of cuts. The number of cells in a particular discrete program may not be as small as the minimum possible value because the program that generated the linear programs may not have detected all empty slab intersections and thus may have included constraints in the program for extraneous cells.

The solutions to the linear programs suggest that for cutting schemes involving a single type of cut, sparcs are more effective than pair-isolating cuts for the purpose of obtaining small approximation ratios. In general, we found that for a fixed grid size N, when only one type of cut is included in the program, an optimal solution involving sparcs has a smaller maximum density than an optimal solution involving either type of pair-isolating cut. When sparcs are combined with either type of pair-isolating cut, however, the resulting linear program has a smaller optimal value than the optimal value of the program that contains only sparcs. This suggests that pair-isolating cuts, when introduced in addition to sparcs, are useful in improving the best approximation ratio that can be achieved.

A logical postulate is that an optimal cutting scheme for $k = 4$ cannot be described as a probability distribution over sparcs. It is possible, however, that pair-isolating cuts improve the maximum density when introduced into a program containing sparcs only because the linear program is a discrete version of an infinite-dimensional program. For discrete linear programs including only pair-side cuts, the probability that the optimal solution assigns to the sparcs increases as the grid size N increases. It is conceivable that pair-isolating cuts only improve the approximation ratio for small grid sizes, and, as the grid size increases so that the discrete program better approximates the infinite-dimensional continuous program, the total probability assigned to the pair-isolating cuts decreases, reaching zero in the limit.

Table 4.2 shows the best approximation ratios we found for various combinations of types of cuts by solving the discrete primal linear programs.

Cut types	Grid	Max. density	sparc	pair-side	pair-edge
pair-side	31	1.176081	N/A	1	N/A
pair-edge	18	1.212101	N/A	N/A	1
pair-side, pair-edge	16	1.189131	N/A	0.999878	0.000122
sparc, pair-side	30	1.15683	0.966739	0.033262	N/A
sparc, pair-edge	16	1.172248	0.877365	N/A	0.122635
sparc, pair-side, pair-edge	15	1.172152	0.743059	0.228798	0.028143

Table 4.2: The maximum densities of the cutting schemes specified by the optimal solutions to the discrete primal linear programs for various combinations of types of cuts.

Of the cutting schemes that we found as solutions to the primal linear programs, the one that achieves the best approximation ratio when used as the rounding scheme in the Călinescu et al. algorithm is a probability distribution over sparcs and pair-side cuts. This may be the result of the fact that we were able to solve the discrete linear programs including sparcs and pair-side cuts for higher grid sizes than the programs including sparcs and pair-edge cuts. Our results showed, however, that for a fixed grid size N, an optimal solution to a program with sparcs and pair-side cuts would lead to a smaller approximation ratio than an optimal solution to a program with sparcs and pair-edge cuts.

4.3 Observations

In general, the solutions to the discrete linear programs exhibit little obvious structure. Detecting patterns in the probability distributions over 4-way cuts of the simplex is not a straightforward task. Because a solution to the primal linear program is a joint distribution over cuts of the various types included in the program, examining the solutions themselves does not provide immediate insight into the properties of the slices that comprise the cuts. An understanding of the distributions of the various component slices would be a natural approach to take in an attempt to extrapolate from the solutions to the discrete linear programs to an analysis of a cutting scheme that demonstrates an approximation ratio close to the objective values of the solutions. While we computed marginal distributions of the component slices over the slice slabs, and conditional distributions of the slices, given the slabs in which the other slices fall, we did not make significant progress in the use of these distributions to develop insight into the properties of the component slices of the cuts.

Corner cuts appear to diminish in structure when pair-isolating side-parallel cuts are added to a discrete linear program containing sparcs, especially for higher grid sizes. Karger et al. note that in the optimal solutions to the programs including only sparcs, the slice distances in a corner cut are all chosen uniformly at random from the corner region. In general, optimal solutions to the programs that contain both sparcs and pair-side cuts show that when one of the slice distances is in the corner near $\rho_i = 1$, then the other slice distances are likely to also be in the corner. However, the slice distances do not appear to be distributed uniformly over the corner slabs. On the other hand, in the case of the discrete linear programs including sparcs and pair-edge cuts, corner cuts appear to remain intact, with the slice distances distributed uniformly over the corner slabs.

The dual of the discrete linear program that we solved to obtain a cutting scheme contains one variable for each cell in the simplex. We may consider the value of the variable for a cell to be a cost assigned to each segment in the cell. Intuitively, the objective of the dual program is to assign a fixed amount of cost to the cells such that any k-way cut of the simplex will have a high cost. In this sense, a solution to the dual program identifies regions in the simplex that contain segments on which any cutting scheme must have high density. While solutions to the dual linear programs may offer insight into how the density of an optimal cutting scheme varies in different regions of the simplex, we did not find a way to exploit this information in the development and analysis of cutting schemes that achieve small approximation ratios. We had difficulty in visualizing solutions to the dual programs, as the task of representing the cells in a visual form such that one may understand how the costs are assigned to the different cells is formidable.

To obtain a simple visual representation of the regions of high and low density in the simplex for a certain cutting scheme, we worked with another discrete primal linear program in addition to the one that has constraints for cells in the simplex. In the alternate program, rather than considering regions of the simplex for computing density bounds, we instead focus on the aligned line segments in the simplex of length $1/N$ whose endpoints are of the form $(q_1/N, q_2/N, q_3/N, q_4/N)$ for integers $q_i \in [0, N-1]$. The variables of the program remain the discrete 4-way cuts of the simplex, but we replace the constraints for the cells with one constraint for each of the line segments.

Each constraint requires that the density of the cutting scheme on the segment be at most the value of the objective function. Minimizing the objective function thus leads to a solution in which the value of the objective function is the maximum density of the cutting scheme specified by the values of the variables for the 4-way cuts of the simplex. Generating the constraint in the program for a particular segment involves computing the density of each discrete cut on the segment. The calculation is similar to the computation of an upper bound on the density of a cut on any segment in a cell, but is less complicated because the density must be computed for a particular known line segment, rather than an arbitrary segment within a cell.

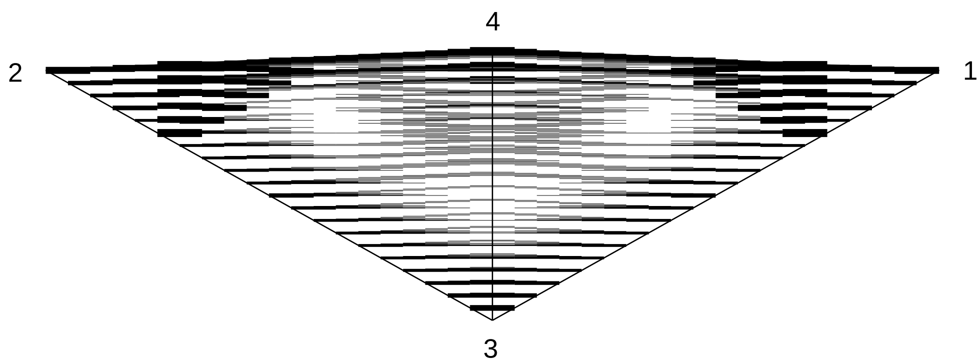
A solution to this linear program is a cutting scheme that has a small maximum density when the set of possible embedded edges is restricted to include only aligned segments of length $1/N$ whose endpoints are of the form $(q_1/N, q_2/N, q_3/N, q_4/N)$. Because the program does not account for every possible embedded edge in the simplex, the objective value of a solution to it does not provide an upper bound on the approximation ratio achieved by the cutting scheme specified by the solution. If the grid size N is large enough, however, we may hope that many graphs have optimal embeddings in which all the edges map to segments accounted for in this program. In this event, a solution to the linear program may illustrate properties of a cutting scheme that has a small maximum density when all segments in the simplex are considered.

When this linear program includes only variables for sparcs, the objective value of an optimal solution to it is a lower bound on the maximum density of any cutting scheme that is a probability distribution over sparcs. To see this, first note that the objective value of a solution is the maximum density of the cutting scheme specified by the solution over the segments for which the program has constraints. Because the set of segments for which the program has constraints is a subset of the set of all segments, the maximum density of the cutting scheme specified by a solution to the program is at least the maximum density of the scheme over the segments for which the program has constraints.

The line segments for which the program has constraints form a particular embedded graph. This embedded graph is constructed such that each embedded edge in it spans precisely one side-parallel slice slab. As such, for any particular slab, any slice chosen from that slab will cut the same set of embedded edges, regardless of the exact slice distance within the slab. It follows that two sparc cutting schemes that have the same distribution over the slice slabs will have the same density on any edge in the embedded graph, even if the conditional distributions of the slice distances within the slabs differ. We may conclude from this that no sparc cutting scheme can have a smaller maximum density over the segments for which the program has constraints than the cutting scheme over continuous sparcs induced by the optimal solution to the linear program. As the maximum density of any cutting scheme over the segments for which the program has constraints is a lower bound on the maximum density of the cutting scheme, it follows that no sparc cutting scheme can have a maximum density less than the objective value of an optimal solution to the program.

Given a solution to this discrete primal linear program with constraints corresponding to segments, we may determine the density of any set of discrete cuts on a particular segment by applying the total probability theorem using the values of the cut variables in the set from the solution and the densities of the cuts on the segments computed for the constraints. Figure 4-4 shows the density of pair-side cuts in a optimal solution to a linear program including sparcs and pair-side cuts on the different $1, 2$-aligned segments in the simplex. In the figure, the higher the density of the cuts on a segment, the thicker and darker the segment appears.

For the pair-side cuts, the regions containing segments on which the cuts have high density include several areas near the edges of the simplex. The edge connecting vertices 3 and 4 of the simplex, however, does not have an area of high density near it. The segments with the highest densities are concentrated near the corners of the simplex closest to vertices 1 and 2.



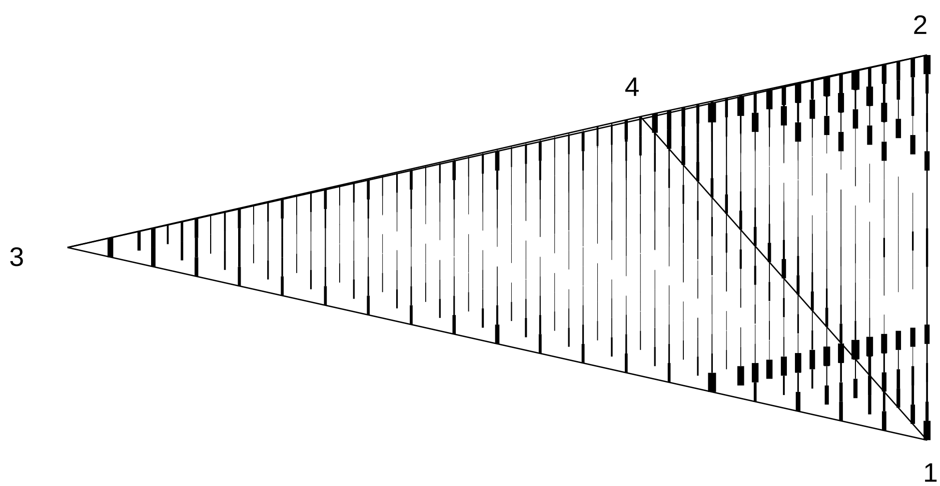


Figure 4-4: The density of the pair-isolating side-parallel cuts on the $1, 2$-aligned segments in the simplex. For each segment, the density of the cuts on the segment is represented by its thickness and darkness.

Chapter 5

Cutting Schemes Involving Dependent Side-Parallel Slices

While the cutting schemes found by Karger et al. that have the smallest maximum densities are probability distributions that they obtained from solutions to the discrete primal linear programs, those cutting schemes have the disadvantage that the proofs of the upper bounds on the approximation ratios they achieve are computer-generated. In their study of the Călinescu et al. relaxation, Karger et al. give a sparc cutting scheme for arbitrary k that achieves approximation ratios that may be proven analytically to be smaller than those of the Călinescu et al. algorithm. This cutting scheme involves a random choice between a corner cut and another cutting scheme in which the slice distance for each side-parallel slice is chosen, independently of the other slice distances, uniformly at random from the portion of the range of the slice distances obtained by removing the corner region from the full range.

For $k = 4$, this cutting scheme achieves the smallest approximation ratio among all known cutting schemes for which upper bounds on the maximum densities have been proven analytically. In this chapter, we present a cutting scheme for the 4-simplex involving side-parallel cuts in which there is a dependence between the slice distances of the different side-parallel slices. Our analysis of the maximum density of this cutting scheme shows that the approximation ratio it achieves improves upon the approximation ratio of the Karger et al. cutting scheme.

5.1 A General Class of Cutting Schemes

To specify the cutting scheme, we first describe a class of sparc cutting schemes for which the distributions of the slice distances of different slices have dependencies. We then give a set of parameters that identify a particular cutting scheme, and compute an upper bound on the maximum density of this cutting scheme.

5.1.1 Description of cutting schemes

Divide the range $[0, 1]$ from which the slice distances are drawn into t slabs, with slab i being the range $[b_i, b_i + w_i]$ of length w_i for $i = 1, \ldots, t$. Note that $b_{i+1} = b_i + w_i$ and $\sum_{i=1}^{t} w_i = 1$. We partition the simplex into four subsets by choosing three slice distances ρ_i, and processing the vertices in the order specified by a permutation σ of the vertices chosen uniformly at random from the set of possible permutations. The ith vertex in the permutation, $\sigma(i)$, is separated from the other vertices by the ith side-parallel slice, $\Delta_{x_{\sigma(i)}=\rho_i}$. As in the case of discrete sparcs, we specify the distribution of a slice distance by providing the slab $r \in \{1, \ldots, t\}$ in which the slice falls. The slice distance is then chosen uniformly at random from the range of values for the slab, $[b_r, b_r + w_r]$.

Let $p_1(i)$ denote the probability that the first slice is in slab i, $p_2(j \mid i)$ be the conditional probability that the second slice is in slab j, given that the first slice is in slab i, and $p_3(\ell \mid i, j)$ be the conditional probability that the third slice is in slab ℓ, given that the first slice is in slab i and the second slice is in slab j. By using conditional distributions in the description of the cutting schemes, we allow for dependencies between the different slice distances. Given these parameters, we may specify a probability distribution over sparcs by defining the slabs and providing the probabilities $p_1(i)$, $p_2(j \mid i)$, and $p_3(\ell \mid i, j)$ for all the slabs.

5.1.2 Density analysis

We obtain an upper bound on the maximum density of a cutting scheme of this form through the same approach that we used to compute the constraints for the discrete linear program. Again, we consider cells of the simplex induced by the slabs for the slice distances. For each cell, we compute an upper bound on the density of the cutting scheme an any segment in the cell. Because the permutation of the vertices used to determine the order in which they are processed is selected uniformly at random from the set of possible permutations, the cutting scheme is symmetric with respect to all coordinates, and so it is sufficient to consider only $1,2$-aligned segments in the computations of the density bounds. Note that any segment that lies in more than one cell may be divided into several parts, each contained entirely in one cell, such that the density bounds for each cell may be applied to the portion of the segment in that cell. The maximum density of the cutting scheme is at most the largest of the upper bounds for the different cells.

Consider the cell in the simplex identified by the slabs (r_1, r_2, r_3, r_4), where r_i denotes the slab containing coordinate i of the points in the cell. For all points x in the cell, then, $x_i \in [b_{r_i}, b_{r_i}+w_{r_i}]$. Since the cell is in the simplex, $\sum_{i=1}^{4} x_i = 1$ for points x in it, and therefore the slab boundaries must satisfy $\sum_{i=1}^{4} b_{r_i} \leq 1 \leq \sum_{i=1}^{4}(b_{r_i}+w_{r_i})$. Note that if either of these inequalities holds with equality, then each coordinate x_i must be precisely equal to the boundary value b_{r_i} or $b_{r_i}+w_{r_i}$ in order for the point x to be in the cell. It follows that there can be only one point in the cell. Because a cell consisting of only one point cannot contain a line segment, we need only consider cells for which $\sum_{i=1}^{4} b_{r_i} < 1 < \sum_{i=1}^{4}(b_{r_i}+w_{r_i})$.

For a $1,2$-aligned segment of length d with endpoints (s_1+d, s_2, s_3, s_4) and (s_1, s_2+d, s_3, s_4) in the cell (r_1, r_2, r_3, r_4), let C denote the number of times the segment is cut by a sparc obtained by choosing the slice distances according to the distributions specified by the parameters of the cutting scheme. We denote the slab that contains the ith slice by u_i. To compute the expected value of C, we consider several cases.

Case 1: $\sigma(4) = 2$

Since a $1,2$-aligned segment can be cut only by terminal 1 or terminal 2, and the last vertex in the permutation does not cut any segments, if the last vertex in the permutation is 2, then the segment can be cut only by terminal 1. The probability that terminal 1 cuts the segment depends on the position of vertex 1 in the permutation.

Case 1A: $\sigma(1) = 1$, $\sigma(4) = 2$

If vertex 1 is the first vertex in the permutation, then the segment may be cut by the first side-parallel slice if the slab containing the slice is r_1. Because the conditional distribution of the slice distance, given that the slice is in a particular slab, is uniform over the slab, in the event that the slab containing the first slice is r_1, the probability that the slice cuts the segment is d/w_{r_1}. As a result, the expected number of times the segment is cut is $E[C \mid \sigma(1) = 1, \sigma(4) = 2] = (p_1(r_1)/w_{r_1})d$ when vertex 1 is the first vertex in the permutation.

Case 1B: $\sigma(2) = 1$, $\sigma(4) = 2$

In the case that vertex 1 is the second vertex in the permutation, the segment can be cut by the second slice only if it is not captured by the terminal whose corresponding vertex is first in the permutation. The segment will not be captured if the slab u_1 in which the first slice falls satisfies $u_1 > r_{\sigma(1)}$, or if $u_1 = r_{\sigma(1)}$ and the slice distance ρ_1 satisfies $\rho_i > s_{\sigma(1)}$. To compute the probability that the segment is cut, we determine the probability that the second slice is in slab r_1, given that the segment is not captured, and we multiply that probability by the probability that the segment is not captured and the probability that the segment is cut, given that it is not captured and the second slice is in slab r_1. Let NC_i be the event that the segment is not captured or cut by the terminal corresponding to the ith vertex in the permutation, $\sigma(i)$. The probability that the slab u_2 containing the second slice is the same as the slab r_1, given that the segment is not captured, may be obtained by conditioning on the slab u_1 of the first slice.

$$
\begin{aligned}
&\Pr[u_2 = r_1 \mid NC_1, \sigma(2) = 1, \sigma(4) = 2] \\
&\quad = \sum_{i=r_{\sigma(1)}+1}^{t} (\Pr[u_2 = r_1 \mid u_1 = i, NC_1, \sigma(2) = 1, \sigma(4) = 2] \\
&\qquad\qquad \times \Pr[u_1 = i \mid NC_1, \sigma(2) = 1, \sigma(4) = 2]) \\
&\qquad + (\Pr[u_2 = r_1 \mid u_1 = r_{\sigma(1)}, NC_1, \sigma(2) = 1, \sigma(4) = 2] \\
&\qquad\qquad \times \Pr[u_1 = r_{\sigma(1)} \mid NC_1, \sigma(2) = 1, \sigma(4) = 2])
\end{aligned}
$$

For $i > r_{\sigma(1)}$, the probability that the first slice is in slab i, given that the segment is not captured, is $\Pr[u_1 = i \mid NC_1, \sigma(2) = 1, \sigma(4) = 2] = \Pr[u_1 = i, NC_1 \mid \sigma(2) = 1, \sigma(4) = 2]/\Pr[NC_1 \mid \sigma(2) = 1, \sigma(4) = 2] = p_1(i)/\Pr[NC_1 \mid \sigma(2) = 1, \sigma(4) = 2]$. If the first slice is in slab $r_{\sigma(1)}$, then, since the slice distance is distributed uniformly over $[b_{r_{\sigma(1)}}, b_{r_{\sigma(1)}}+w_{r_{\sigma(1)}}]$, the probability that the segment is not captured is $(b_{r_{\sigma(1)}}+w_{r_{\sigma(1)}}-s_{\sigma(1)})/w_{r_{\sigma(1)}}$. For any coordinate i, let $y_i = b_{r_i}+w_{r_i}-s_i$. The probability that the first slice is in slab $r_{\sigma(1)}$, given that the segment is not captured, is then given by $\Pr[u_1 = r_{\sigma(1)} \mid NC_1, \sigma(2) = 1, \sigma(4) = 2] = \Pr[u_1 = r_{\sigma(1)}, NC_1 \mid \sigma(2) = 1, \sigma(4) = 2]/\Pr[NC_1 \mid \sigma(2) = 1, \sigma(4) = 2] = (p_1(r_{\sigma(1)})y_{\sigma(1)})/(w_{r_{\sigma(1)}}\Pr[NC_1 \mid \sigma(2) = 1, \sigma(4) = 2])$. Using these probabilities and the fact that $\Pr[u_2 = r_1 \mid u_1 = i, NC_1, \sigma(2) = 1, \sigma(4) = 2] = p_2(r_1 \mid i)$ for all $i \geq r_{\sigma(1)}$, we may compute $\Pr[u_2 = r_1 \mid NC_1, \sigma(2) = 1, \sigma(4) = 2]$. This allows us to calculate the expected number of times the segment is cut in this case.

$$
\begin{aligned}
&E[C \mid \sigma(2) = 1, \sigma(4) = 2] \\
&\quad = \Pr[NC_1 \mid \sigma(2) = 1, \sigma(4) = 2] \times \Pr[u_2 = r_1 \mid NC_1, \sigma(2) = 1, \sigma(4) = 2] \\
&\qquad \times E[C \mid u_2 = r_1, NC_1, \sigma(2) = 1, \sigma(4) = 2] \\
&\quad = \Pr[NC_1 \mid \sigma(2) = 1, \sigma(4) = 2]\Big(\sum_{i=r_{\sigma(1)}+1}^{t} \frac{p_1(i)p_2(r_1 \mid i)}{\Pr[NC_1 \mid \sigma(2) = 1, \sigma(4) = 2]} \\
&\qquad\qquad + \frac{p_1(r_{\sigma(1)})p_2(r_1 \mid r_{\sigma(1)})y_{\sigma(1)}}{w_{r_{\sigma(1)}}\Pr[NC_1 \mid \sigma(2) = 1, \sigma(4) = 2]}\Big)\frac{d}{w_{r_1}} \\
&\quad = \frac{d}{w_{r_1}}\Big(\sum_{i=r_{\sigma(1)}+1}^{t} p_1(i)p_2(r_1 \mid i) + \frac{p_1(r_{\sigma(1)})p_2(r_1 \mid r_{\sigma(1)})y_{\sigma(1)}}{w_{r_{\sigma(1)}}}\Big)
\end{aligned}
$$

Given that vertex 2 is the last vertex in the permutation and vertex 1 is the second, the first vertex in the permutation is vertex 3 with probability 1/2 and vertex 4 with probability 1/2. Thus, the expected number of times the segment is cut when vertex 1 is the second vertex may be expressed as follows.

$$
\begin{aligned}
&E[C \mid \sigma(2) = 1, \sigma(4) = 2] \\
&\quad = \frac{d}{2w_{r_1}} \Big(\sum_{i=r_3+1}^{t} p_1(i) p_2(r_1 \mid i) + \frac{p_1(r_3) p_2(r_1 \mid r_3) y_3}{w_{r_3}} \\
&\qquad\qquad + \sum_{i=r_4+1}^{t} p_1(i) p_2(r_1 \mid i) + \frac{p_1(r_4) p_2(r_1 \mid r_4) y_4}{w_{r_4}} \Big)
\end{aligned}
$$

Case 1C: $\sigma(3) = 1$, $\sigma(4) = 2$

If vertex 1 is the third vertex in the permutation, then we calculate the expected number of cuts of the segment in a manner analogous to that of the case that vertex 1 is the second vertex. The segment can be cut only if neither of the terminals corresponding to the first two vertices in the permutation captures it. Let NC_{12} be the event that the segment is not captured or cut by either of the terminals corresponding to the first two vertices in the permutation. The event NC_{12} can occur only if $u_1 \geq r_{\sigma(1)}$ and $u_2 \geq r_{\sigma(2)}$. We condition on these values of u_1 and u_2 to compute the expected number of times the segment is cut.

$$
\begin{aligned}
&E[C \mid \sigma(3) = 1, \sigma(4) = 2] \\
&\quad = \Pr[NC_{12} \mid \sigma(3) = 1, \sigma(4) = 2] \times \Pr[u_3 = r_1 \mid NC_{12}, \sigma(3) = 1, \sigma(4) = 2] \\
&\qquad \times E[C \mid u_3 = r_1, NC_{12}, \sigma(3) = 1, \sigma(4) = 2]
\end{aligned}
$$

$$
\begin{aligned}
&\Pr[u_3 = r_1 \mid NC_{12}, \sigma(3) = 1, \sigma(4) = 2] \\
&\quad = \sum_{i=r_{\sigma(1)}}^{t} \sum_{j=r_{\sigma(2)}}^{t} (\Pr[u_3 = r_1 \mid u_1 = i, u_2 = j, NC_{12}, \sigma(3) = 1, \sigma(4) = 2] \\
&\qquad\qquad \times \Pr[u_1 = i, u_2 = j \mid NC_{12}, \sigma(3) = 1, \sigma(4) = 2]) \\
&\quad = \sum_{i=r_{\sigma(1)}+1}^{t} \sum_{j=r_{\sigma(2)}+1}^{t} \frac{p_1(i) p_2(j \mid i) p_3(r_1 \mid i, j)}{\Pr[NC_{12} \mid \sigma(3) = 1, \sigma(4) = 2]}
\end{aligned}
$$

$$+\sum_{i=r_{\sigma(1)}+1}^{t}\frac{p_1(i)p_2(r_{\sigma(2)}\mid i)p_3(r_1\mid i,r_{\sigma(2)})y_{\sigma(2)}}{w_{r_{\sigma(2)}}\Pr[NC_{12}\mid\sigma(3)=1,\sigma(4)=2]}$$
$$+\sum_{j=r_{\sigma(2)}+1}^{t}\frac{p_1(r_{\sigma(1)})p_2(j\mid r_{\sigma(1)})p_3(r_1\mid r_{\sigma(1)},j)y_{\sigma(1)}}{w_{r_{\sigma(1)}}\Pr[NC_{12}\mid\sigma(3)=1,\sigma(4)=2]}$$
$$+\frac{p_1(r_{\sigma(1)})p_2(r_{\sigma(2)}\mid r_{\sigma(1)})p_3(r_1\mid r_{\sigma(1)},r_{\sigma(2)})y_{\sigma(1)}y_{\sigma(2)}}{w_{r_{\sigma(1)}}w_{r_{\sigma(2)}}\Pr[NC_{12}\mid\sigma(3)=1,\sigma(4)=2]}$$

$$\begin{aligned}
&E[C\mid\sigma(3)=1,\sigma(4)=2]\\
&=\frac{d}{2w_{r_1}}\Big(\sum_{i=r_3+1}^{t}\sum_{j=r_4+1}^{t}p_1(i)p_2(j\mid i)p_3(r_1\mid i,j)+\sum_{i=r_4+1}^{t}\sum_{j=r_3+1}^{t}p_1(i)p_2(j\mid i)p_3(r_1\mid i,j)\\
&\quad+\sum_{i=r_3+1}^{t}\frac{p_1(i)p_2(r_4\mid i)p_3(r_1\mid i,r_4)y_4}{w_{r_4}}+\sum_{i=r_4+1}^{t}\frac{p_1(i)p_2(r_3\mid i)p_3(r_1\mid i,r_3)y_3}{w_{r_3}}\\
&\quad+\sum_{j=r_4+1}^{t}\frac{p_1(r_3)p_2(j\mid r_3)p_3(r_1\mid r_3,j)y_3}{w_{r_3}}\\
&\quad+\sum_{j=r_3+1}^{t}\frac{p_1(r_4)p_2(j\mid r_4)p_3(r_1\mid r_4,j)y_4}{w_{r_4}}\\
&\quad+\frac{(p_1(r_3)p_2(r_4\mid r_3)p_3(r_1\mid r_3,r_4)+p_1(r_4)p_2(r_3\mid r_4)p_3(r_1\mid r_4,r_3))y_3y_4}{w_{r_3}w_{r_4}}\Big)
\end{aligned}$$

Conditioning on position of vertex 1 in permutation

We may now compute the expected number of times the segment is cut in the case that vertex 2 is the last vertex in the permutation by noting that, given that vertex 2 is last, vertex 1 is equally likely to be first, second, and third in the permutation.

$$\begin{aligned}
&E[C\mid\sigma(4)=2]=\\
&\frac{1}{3}(E[C\mid\sigma(1)=1,\sigma(4)=2]+E[C\mid\sigma(2)=1,\sigma(4)=2]\\
&\quad+E[C\mid\sigma(3)=1,\sigma(4)=2])
\end{aligned}$$

Case 2: $\sigma(4)=1$

The case in which vertex 1 is the last vertex in the permutation is symmetric to the case in which vertex 2 is the last vertex. Each of these cases occurs with probability 1/4.

Case 3: $\sigma(4) = 4$

Because both vertices 1 and 2 are within the first three vertices of the permutation, the segment may be cut by both terminals 1 and 2. The calculation of the expected number of times the segment is cut, conditional on the last vertex in the permutation being vertex 4, is similar to the calculation for the case of vertex 2 as the last vertex, though we must now account for the possibility that each of two terminals cuts the segment.

Case 3A: $\sigma(1) = 1$, $\sigma(2) = 2$, $\sigma(4) = 4$

Let CUT_i be the event that the segment is cut by the terminal corresponding to the ith vertex in the permutation. Terminal 1 cuts the segment with probability $\Pr[CUT_1 \mid \sigma(1) = 1, \sigma(2) = 2, \sigma(4) = 4] = (p_1(r_1)/w_{r_1})d$. If terminal 1 cuts the segment, then it will capture part of the segment that terminal 2 cannot subsequently cut.

Let γ be the fraction of the segment not captured by terminal 1, so that $\rho_1 = s_1 + \gamma d$. Since $x_1+x_2 = s_1+s_2+d$ for all points x on the segment, the points on the segment that are not captured by terminal 1 are those for which $x_1 = -x_2+s_1+s_2+d \leq s_1+\gamma d$, or $x_2 \geq s_2+(1-\gamma)d$. Thus, terminal 2 cuts the segment if $s_2+(1-\gamma)d \leq \rho_2 \leq s_2+d$, and the probability that the segment is cut by terminal 2, given that it is cut by terminal 1, is $(p_2(r_2 \mid r_1)(d-(1-\gamma)d))/w_{r_2} = (p_2(r_2 \mid r_1)\gamma d)/w_{r_2}$. Because the slice distance ρ_1 is distributed uniformly throughout the slab r_1, given that the segment is cut by terminal 1, γ is distributed uniformly over $[0,1]$, and therefore it has an expected value of 1/2. It follows that the expected number of times the segment is cut, conditional on the event that terminal 1 cuts it, is $1 + (p_2(r_2 \mid r_1)/(2w_{r_2}))d$.

In the event that the segment is not cut by terminal 1, it can be cut by terminal 2 only if it is not captured by terminal 1. Terminal 1 will not capture it if $u_1 > r_1$, or if $u_1 = r_1$ and $\rho_1 > s_1 + d$. If $u_1 = r_1$, then the probability that terminal 1 does not capture the segment is $(b_{r_1} + w_{r_1} - s_1 - d)/w_{r_1} = (y_1 - d)/w_{r_1}$. We condition on the different values of u_1 to determine an upper bound on the expected number of times the segment is cut.

$$
\begin{aligned}
&E[C \mid \sigma(1)=1, \sigma(2)=2, \sigma(4)=4] \\
&\quad = \Pr[CUT_1 \mid \sigma(1)=1, \sigma(2)=2, \sigma(4)=4] \\
&\qquad \times E[C \mid CUT_1, \sigma(1)=1, \sigma(2)=2, \sigma(4)=4] \\
&\qquad + \Pr[NC_1 \mid \sigma(1)=1, \sigma(2)=2, \sigma(4)=4] \\
&\qquad\quad \times E[C \mid NC_1, \sigma(1)=1, \sigma(2)=2, \sigma(4)=4] \\
&\quad = \frac{p_1(r_1)d}{w_{r_1}}\left(1+\frac{p_2(r_2 \mid r_1)}{2w_{r_2}}d\right) \\
&\qquad + \Bigl(\Pr[NC_1 \mid \sigma(1)=1, \sigma(2)=2, \sigma(4)=4] \\
&\qquad\quad \times \Bigl(\sum_{i=r_1+1}^{t} \frac{p_1(i)p_2(r_2 \mid i)}{\Pr[NC_1 \mid \sigma(1)=1, \sigma(2)=2, \sigma(4)=4]} \\
&\qquad\qquad + \frac{p_1(r_1)p_2(r_2 \mid r_1)(y_1-d)}{w_{r_1}\Pr[NC_1 \mid \sigma(1)=1, \sigma(2)=2, \sigma(4)=4]}\Bigr)\frac{d}{w_{r_2}}\Bigr) \\
&\quad = \frac{p_1(r_1)d}{w_{r_1}} + \frac{d}{w_{r_2}}\Bigl(\sum_{i=r_1+1}^{t} p_1(i)p_2(r_2 \mid i) + \frac{p_1(r_1)p_2(r_2 \mid r_1)(y_1-\frac{1}{2}d)}{w_{r_1}}\Bigr)
\end{aligned}
$$

Case 3B: $\sigma(1)=2$, $\sigma(2)=1$, $\sigma(4)=4$

The case of $\sigma(1)=2$, $\sigma(2)=1$, and $\sigma(4)=4$ is symmetric to the case of $\sigma(1)=1$, $\sigma(2)=2$, and $\sigma(4)=4$. Thus, the same calculation may be performed to obtain an upper bound on $E[C \mid \sigma(1)=2, \sigma(2)=1, \sigma(4)=4]$.

Case 3C: $\sigma(1)=1$, $\sigma(3)=2$, $\sigma(4)=4$

The computation of the upper bound on the expected value of the number of times the segment is cut is similar to the calculations in the previous cases, with the difference being that we must consider the possibility of the segment being captured by terminal 3. To account for this possibility, we use the same approach that we used for the case in which the last vertex in the permutation is 2.

$$
\begin{aligned}
&E[C \mid \sigma(1)=1,\sigma(3)=2,\sigma(4)=4] \\
&\quad = \Pr[CUT_1 \mid \sigma(1)=1,\sigma(3)=2,\sigma(4)=4] \\
&\qquad \times E[C \mid CUT_1,\sigma(1)=1,\sigma(3)=2,\sigma(4)=4] \\
&\qquad + \Pr[NC_{12} \mid \sigma(1)=1,\sigma(3)=2,\sigma(4)=4] \\
&\qquad\quad \times E[C \mid NC_{12},\sigma(1)=1,\sigma(3)=2,\sigma(4)=4] \\
&\quad = \frac{p_1(r_1)d}{w_{r_1}}(1+\Pr[NC_2 \mid CUT_1,\sigma(1)=1,\sigma(3)=2,\sigma(4)=4] \\
&\qquad\qquad\qquad \times E[C \mid NC_2, CUT_1,\sigma(1)=1,\sigma(3)=2,\sigma(4)=4]) \\
&\qquad + \Pr[NC_{12} \mid \sigma(1)=1,\sigma(3)=2,\sigma(4)=4] \\
&\qquad\quad \times \Big(\sum_{i=r_1}^{t}\sum_{j=r_3}^{t}(\Pr[u_3=r_2 \mid u_1=i,u_2=j,NC_{12},\sigma(1)=1,\sigma(3)=2,\sigma(4)=4] \\
&\qquad\qquad\qquad \times \Pr[u_1=i,u_2=j \mid NC_{12},\sigma(1)=1,\sigma(3)=2,\sigma(4)=4])\Big) \\
&\qquad\quad \times E[C \mid u_3=r_2,NC_{12},\sigma(1)=1,\sigma(3)=2,\sigma(4)=4] \\
&\quad = \frac{p_1(r_1)d}{w_{r_1}} \\
&\qquad \times (1+\Pr[NC_2 \mid CUT_1,\sigma(1)=1,\sigma(3)=2,\sigma(4)=4] \\
&\qquad\qquad \times \Big(\sum_{j=r_3}^{t}(\Pr[u_3=r_2 \mid u_2=j,NC_2,CUT_1,\sigma(1)=1,\sigma(3)=2,\sigma(4)=4] \\
&\qquad\qquad\qquad \times \Pr[u_2=j \mid NC_2,CUT_1,\sigma(1)=1,\sigma(3)=2,\sigma(4)=4])\Big) \\
&\qquad\qquad \times E[C \mid u_3=r_2 \mid NC_2,CUT_1,\sigma(1)=1,\sigma(3)=2,\sigma(4)=4]) \\
&\qquad + \Pr[NC_{12} \mid \sigma(1)=1,\sigma(3)=2,\sigma(4)=4] \\
&\qquad\quad \times \Big(\sum_{i=r_1+1}^{t}\sum_{j=r_3+1}^{t}\frac{p_1(i)p_2(j \mid i)p_3(r_2 \mid i,j)}{\Pr[NC_{12} \mid \sigma(1)=1,\sigma(3)=2,\sigma(4)=4]} \\
&\qquad\qquad + \sum_{i=r_1+1}^{t}\frac{p_1(i)p_2(r_3 \mid i)p_3(r_2 \mid i,r_3)y_3}{w_{r_3}\Pr[NC_{12} \mid \sigma(1)=1,\sigma(3)=2,\sigma(4)=4]} \\
&\qquad\qquad + \sum_{j=r_3+1}^{t}\frac{p_1(r_1)p_2(j \mid r_1)p_3(r_2 \mid r_1,j)(y_1-d)}{w_{r_1}\Pr[NC_{12} \mid \sigma(1)=1,\sigma(3)=2,\sigma(4)=4]}
\end{aligned}
$$

$$
\begin{aligned}
&\quad + \frac{p_1(r_1)p_2(r_3 \mid r_1)p_3(r_2 \mid r_1, r_3)(y_1 - d)y_3}{w_{r_1} w_{r_3} \Pr[NC_{12} \mid \sigma(1) = 1, \sigma(3) = 2, \sigma(4) = 4]}\Big)\frac{d}{w_{r_2}} \\
&= \frac{p_1(r_1)d}{w_{r_1}}(1 + \Pr[NC_2 \mid CUT_1, \sigma(1) = 1, \sigma(3) = 2, \sigma(4) = 4] \\
&\qquad \times \Big(\sum_{j=r_3+1}^{t} \frac{p_2(j \mid r_1)p_3(r_2 \mid r_1, j)}{\Pr[NC_2 \mid CUT_1, \sigma(1) = 1, \sigma(3) = 2, \sigma(4) = 4]} \\
&\qquad\quad + \frac{p_2(r_3 \mid r_1)p_3(r_2 \mid r_1, r_3)y_3}{w_{r_3}\Pr[NC_2 \mid CUT_1, \sigma(1) = 1, \sigma(3) = 2, \sigma(4) = 4]}\Big)\frac{d}{2w_{r_2}} \\
&\quad + \frac{d}{w_{r_2}}\Big(\sum_{i=r_1+1}^{t} \sum_{j=r_3+1}^{t} p_1(i)p_2(j \mid i)p_3(r_2 \mid i, j) \\
&\qquad + \sum_{i=r_1+1}^{t} \frac{p_1(i)p_2(r_3 \mid i)p_3(r_2 \mid i, r_3)y_3}{w_{r_3}} \\
&\qquad + \sum_{j=r_3+1}^{t} \frac{p_1(r_1)p_2(j \mid r_1)p_3(r_2 \mid r_1, j)(y_1 - d)}{w_{r_1}} \\
&\qquad + \frac{p_1(r_1)p_2(r_3 \mid r_1)p_3(r_2 \mid r_1, r_3)(y_1 - d)y_3}{w_{r_1} w_{r_3}}\Big) \\
&= \frac{p_1(r_1)d}{w_{r_1}} + \frac{d}{w_{r_2}}\Big(\sum_{i=r_1+1}^{t} \sum_{j=r_3+1}^{t} p_1(i)p_2(j \mid i)p_3(r_2 \mid i, j) \\
&\qquad + \sum_{i=r_1+1}^{t} \frac{p_1(i)p_2(r_3 \mid i)p_3(r_2 \mid i, r_3)y_3}{w_{r_3}} \\
&\qquad + \sum_{j=r_3+1}^{t} \frac{p_1(r_1)p_2(j \mid r_1)p_3(r_2 \mid r_1, j)(y_1 - \frac{1}{2}d)}{w_{r_1}} \\
&\qquad + \frac{p_1(r_1)p_2(r_3 \mid r_1)p_3(r_2 \mid r_1, r_3)(y_1 - \frac{1}{2}d)y_3}{w_{r_1} w_{r_3}}\Big)
\end{aligned}
$$

Case 3D: $\sigma(1) = 2$, $\sigma(3) = 1$, $\sigma(4) = 4$

This case is symmetric to the case $\sigma(1) = 1$, $\sigma(3) = 2$, $\sigma(4) = 4$.

Case 3E: $\sigma(2) = 1$, $\sigma(3) = 2$, $\sigma(4) = 4$

Again, we compute the expected number of times the segment is cut by terminals 1 and 2, taking into consideration the possibility of terminal 3 capturing the segment.

$$E[C \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$= \Pr[CUT_2, NC_1 \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$\times E[C \mid CUT_2, NC_1, \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$+ \Pr[NC_{12} \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$\times E[C \mid NC_{12}, \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$= \Pr[CUT_2, NC_1 \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$\times (1 + \Big(\sum_{i=r_3}^{t} (\Pr[u_3 = r_2 \mid u_1 = i, CUT_2, NC_1, \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$\times \Pr[u_1 = i \mid CUT_2, NC_1, \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]) \Big)$$

$$\times E[C \mid u_3 = r_2, CUT_2, NC_1, \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4])$$

$$+ \Pr[NC_{12} \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$\times \Big(\sum_{i=r_3}^{t} \sum_{j=r_1}^{t} (\Pr[u_3 = r_2 \mid u_1 = i, u_2 = j, NC_{12}, \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$\times \Pr[u_1 = i, u_2 = j \mid NC_{12}, \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]) \Big)$$

$$\times E[C \mid u_3 = r_2, NC_{12}, \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$= \Pr[CUT_2, NC_1 \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$\times (1 + \Big(\sum_{i=r_3+1}^{t} \frac{p_1(i) p_2(r_1 \mid i) p_3(r_2 \mid i, r_1) d}{w_{r_1} \Pr[CUT_2, NC_1 \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]}$$

$$+ \frac{p_1(r_3) p_2(r_1 \mid r_3) p_3(r_2 \mid r_3, r_1) y_3 d}{w_{r_1} w_{r_3} \Pr[CUT_2, NC_1 \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]} \Big) \frac{d}{2 w_{r_2}})$$

$$+ \Pr[NC_{12} \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$\times \Big(\sum_{i=r_3+1}^{t} \sum_{j=r_1+1}^{t} \frac{p_1(i) p_2(j \mid i) p_3(r_2 \mid i, j)}{\Pr[NC_{12} \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]}$$

$$+ \sum_{i=r_3+1}^{t} \frac{p_1(i) p_2(r_1 \mid i) p_3(r_2 \mid i, r_1)(y_1 - d)}{w_{r_1} \Pr[NC_{12} \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]}$$

$$+ \sum_{j=r_1+1}^{t} \frac{p_1(r_3) p_2(j \mid r_3) p_3(r_2 \mid r_3, j) y_3}{w_{r_3} \Pr[NC_{12} \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]}$$

$$+ \frac{p_1(r_3) p_2(r_1 \mid r_3) p_3(r_2 \mid r_3, r_1)(y_1 - d) y_3}{w_{r_1} w_{r_3} \Pr[NC_{12} \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]} \Big) \frac{d}{w_{r_2}}$$

$$= \Pr[NC_1 \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4]$$

$$
\begin{aligned}
&\times \Pr[CUT_2 \mid NC_1, \sigma(2)=1, \sigma(3)=2, \sigma(4)=4] \\
&+ \frac{d}{w_{r_2}} \Big(\sum_{i=r_3+1}^{t} \sum_{j=r_1+1}^{t} p_1(i) p_2(j \mid i) p_3(r_2 \mid i, j) \\
&\qquad + \sum_{i=r_3+1}^{t} \frac{p_1(i) p_2(r_1 \mid i) p_3(r_2 \mid i, r_1)(y_1 - \frac{1}{2}d)}{w_{r_1}} \\
&\qquad + \sum_{j=r_1+1}^{t} \frac{p_1(r_3) p_2(j \mid r_3) p_3(r_2 \mid r_3, j) y_3}{w_{r_3}} \\
&\qquad + \frac{p_1(r_3) p_2(r_1 \mid r_3) p_3(r_2 \mid r_3, r_1)(y_1 - \frac{1}{2}d) y_3}{w_{r_1} w_{r_3}} \Big) \\
= {}& \Pr[NC_1 \mid \sigma(2)=1, \sigma(3)=2, \sigma(4)=4] \\
&\times \Big(\sum_{i=r_3}^{t} (\Pr[CUT_2 \mid u_1 = i, NC_1, \sigma(2)=1, \sigma(3)=2, \sigma(4)=4] \\
&\qquad \times \Pr[u_1 = i \mid NC_1, \sigma(2)=1, \sigma(3)=2, \sigma(4)=4]) \Big) \\
&+ \frac{d}{w_{r_2}} \Big(\sum_{i=r_3+1}^{t} \sum_{j=r_1+1}^{t} p_1(i) p_2(j \mid i) p_3(r_2 \mid i, j) \\
&\qquad + \sum_{i=r_3+1}^{t} \frac{p_1(i) p_2(r_1 \mid i) p_3(r_2 \mid i, r_1)(y_1 - \frac{1}{2}d)}{w_{r_1}} \\
&\qquad + \sum_{j=r_1+1}^{t} \frac{p_1(r_3) p_2(j \mid r_3) p_3(r_2 \mid r_3, j) y_3}{w_{r_3}} \\
&\qquad + \frac{p_1(r_3) p_2(r_1 \mid r_3) p_3(r_2 \mid r_3, r_1)(y_1 - \frac{1}{2}d) y_3}{w_{r_1} w_{r_3}} \Big) \\
= {}& \Pr[NC_1 \mid \sigma(2)=1, \sigma(3)=2, \sigma(4)=4] \\
&\times \Big(\sum_{i=r_3+1}^{t} \frac{p_1(i) p_2(r_1 \mid i) d}{w_{r_1} \Pr[NC_1 \mid \sigma(2)=1, \sigma(3)=2, \sigma(4)=4]} \\
&\qquad + \frac{p_1(r_3) p_2(r_1 \mid r_3) y_3 d}{w_{r_1} w_{r_3} \Pr[NC_1 \mid \sigma(2)=1, \sigma(3)=2, \sigma(4)=4]} \Big) \\
&+ \frac{d}{w_{r_2}} \Big(\sum_{i=r_3+1}^{t} \sum_{j=r_1+1}^{t} p_1(i) p_2(j \mid i) p_3(r_2 \mid i, j) \\
&\qquad + \sum_{i=r_3+1}^{t} \frac{p_1(i) p_2(r_1 \mid i) p_3(r_2 \mid i, r_1)(y_1 - \frac{1}{2}d)}{w_{r_1}}
\end{aligned}
$$

$$
\begin{aligned}
&\qquad + \sum_{j=r_1+1}^{t} \frac{p_1(r_3)p_2(j \mid r_3)p_3(r_2 \mid r_3, j)y_3}{w_{r_3}} \\
&\qquad + \frac{p_1(r_3)p_2(r_1 \mid r_3)p_3(r_2 \mid r_3, r_1)(y_1 - \frac{1}{2}d)y_3}{w_{r_1}w_{r_3}}\Big) \\
&= \frac{d}{w_{r_1}}\Big(\sum_{i=r_3+1}^{t} p_1(i)p_2(r_1 \mid i) + \frac{p_1(r_3)p_2(r_1 \mid r_3)y_3}{w_{r_3}}\Big) \\
&\quad + \frac{d}{w_{r_2}}\Big(\sum_{i=r_3+1}^{t} \sum_{j=r_1+1}^{t} p_1(i)p_2(j \mid i)p_3(r_2 \mid i, j) \\
&\qquad + \sum_{i=r_3+1}^{t} \frac{p_1(i)p_2(r_1 \mid i)p_3(r_2 \mid i, r_1)(y_1 - \frac{1}{2}d)}{w_{r_1}} \\
&\qquad + \sum_{j=r_1+1}^{t} \frac{p_1(r_3)p_2(j \mid r_3)p_3(r_2 \mid r_3, j)y_3}{w_{r_3}} \\
&\qquad + \frac{p_1(r_3)p_2(r_1 \mid r_3)p_3(r_2 \mid r_3, r_1)(y_1 - \frac{1}{2}d)y_3}{w_{r_1}w_{r_3}}\Big)
\end{aligned}
$$

Case 3F: $\sigma(2) = 2$, $\sigma(3) = 1$, $\sigma(4) = 4$

This case is symmetric to the case $\sigma(2) = 1$, $\sigma(3) = 2$, $\sigma(4) = 4$.

Conditioning on permutation of vertices

Given the expected number of times the segment is cut for each of these permutations, we now compute an upper bound on the expected number of times the segment is cut when $\sigma(4) = 4$ by applying the total probability theorem, using the fact that each permutation is equally likely to be chosen.

$$
\begin{aligned}
&E[C \mid \sigma(4) = 4] \\
&\quad = \frac{1}{6}(E[C \mid \sigma(1) = 1, \sigma(2) = 2, \sigma(4) = 4] + E[C \mid \sigma(1) = 2, \sigma(2) = 1, \sigma(4) = 4] \\
&\qquad + E[C \mid \sigma(1) = 1, \sigma(3) = 2, \sigma(4) = 4] + E[C \mid \sigma(1) = 2, \sigma(3) = 1, \sigma(4) = 4] \\
&\qquad + E[C \mid \sigma(2) = 1, \sigma(3) = 2, \sigma(4) = 4] + E[C \mid \sigma(2) = 2, \sigma(3) = 1, \sigma(4) = 4])
\end{aligned}
$$

Case 4: $\sigma(4) = 3$

This case is symmetric to the case in which the last vertex in the permutation is vertex 4. The probability that each case occurs is 1/4.

Conditioning on last vertex in permutation

We apply the total probability theorem again to compute an upper bound on the expected number of times the segment is cut over all permutations of the vertices. This allows us to obtain an upper bound on the density $\delta(r_1, r_2, r_3, r_4)$ of the cutting scheme on any 1, 2-aligned segment in the cell.

$$E[C] = \frac{1}{4}(E[C \mid \sigma(4) = 1] + E[C \mid \sigma(4) = 2] + E[C \mid \sigma(4) = 3] + E[C \mid \sigma(4) = 4])$$

$$\delta(r_1, r_2, r_3, r_4) = \frac{E[C]}{d}$$

The maximum density of the cutting scheme is at most the largest of the upper bounds on the density of the cutting scheme for all the cells in the simplex.

5.2 Improved Approximation Ratio

We present a cutting scheme P_d that achieves a smaller approximation ratio than the 1.189 of the Karger et al. cutting scheme for arbitrary k. This cutting scheme is a member of the class of cutting schemes described above, with the following parameters.

For the cutting scheme, there are $t = 3$ slabs. The boundaries of the slabs are $b_1 = 0$, $b_2 = 2/5$, and $b_3 = 3/5$, so that the lengths of the slabs are $w_1 = 2/5$, $w_2 = 1/5$, and $w_3 = 2/5$. The conditional distributions of the slabs in which the various slices fall are as follows.

$$p_1(u_1) = \begin{cases} \frac{1343}{2000}, & u_1 = 1 \\ 0, & u_1 = 2 \\ \frac{657}{2000}, & u_1 = 3 \end{cases} \qquad p_2(u_2 \mid u_1 = 1) = \begin{cases} \frac{21}{100}, & u_2 = 1 \\ \frac{79}{100}, & u_2 = 2 \\ 0, & u_2 = 3 \end{cases}$$

$$p_2(u_2 \mid u_1 = 3) = \begin{cases} 0, & u_2 = 1 \\ 0, & u_2 = 2 \\ 1, & u_2 = 3 \end{cases} \qquad p_3(u_3 \mid u_1 = 1, u_2 = 1) = \begin{cases} 0, & u_3 = 1 \\ 1, & u_3 = 2 \\ 0, & u_3 = 3 \end{cases}$$

$$p_3(u_3 \mid u_1 = 1, u_2 = 2) = \begin{cases} \frac{37}{50}, & u_3 = 1 \\ \frac{13}{50}, & u_3 = 2 \\ 0, & u_3 = 3 \end{cases} \qquad p_3(u_3 \mid u_1 = 3, u_2 = 3) = \begin{cases} 0, & u_3 = 1 \\ 0, & u_3 = 2 \\ 1, & u_3 = 3 \end{cases}$$

We show the following upper bound on the maximum density of this cutting scheme. This result represents an improvement of approximately 0.013 over the Karger et al. bound of 1.189 in the approximation ratio achieved by the rounding scheme in the Călinescu et al. algorithm.

Theorem 5.1. *The maximum density of P_d is at most* 1.1762*.*

Note that because, for a $1,2$-aligned segment, coordinates 1 and 2 are symmetric and coordinates 3 and 4 are symmetric, we need not consider all the cells of the simplex. For the different cells of the simplex, when we compute the upper bounds on the density of the cutting scheme on any segment within the various cells, we obtain the following values.

$$\delta(1,1,1,1) \leq \frac{1343}{1600} + \frac{5335759}{192000000}(y_1 + y_2 - d) + \frac{5335759}{9600000}(y_3 + y_4)$$

$$\delta(1,1,1,2) \le \frac{1343}{1600} + \frac{6745889}{38400000}(y_1+y_2) + \frac{6745889}{19200000}y_3 + \frac{3925589}{7680000}(y_1+y_2-d)y_4 + \frac{3925589}{3840000}y_3y_4$$

$$\delta(1,1,1,3) \le \frac{1343}{1600} + \frac{6745889}{38400000}(y_1+y_2-d) + \frac{6745889}{19200000}y_3$$

$$\delta(1,1,2,2) \le \frac{1343}{1600} + \frac{9401}{128000}(y_1+y_2-d) + \frac{3925589}{7680000}(y_1+y_2-d)(y_3+y_4)$$

$$\delta(1,2,1,1) \le \frac{1343}{3200} + \frac{2228037}{3200000}(y_1-\frac{d}{2}) + \frac{33482333}{38400000}(y_3+y_4) + \frac{9401}{25600}((y_1-\frac{d}{2})(y_3+y_4)+y_3y_4) + \frac{3925589}{7680000}(y_2-\frac{d}{2})(y_3+y_4)$$

$$\delta(1,2,1,2) \le \frac{1343}{3200} + \frac{11988961}{19200000}(y_1-\frac{d}{2}) + \frac{13399111}{19200000}y_3 + \frac{9401}{25600}(y_1-\frac{d}{2})y_3 + \frac{1379261}{3840000}(y_1-\frac{d}{2})y_4 + \frac{3925589}{7680000}(y_2-\frac{d}{2})y_3 + \frac{2228037}{2560000}y_3y_4$$

$$\delta(2,2,1,1) \le \frac{11988961}{9600000}(y_3+y_4) + \frac{1379261}{3840000}(y_1+y_2-d)(y_3+y_4) + \frac{9401}{12800}y_3y_4$$

$$\delta(1,3,1,1) \le \frac{1657}{1600} + \frac{6745889}{38400000}(y_3+y_4)$$

Since $y_i = b_{r_i} + w_{r_i} - s_i$, $\sum_{i=1}^4 y_i = \sum_{i=1}^4 (b_{r_i} + w_{r_i} - s_i) = \sum_{i=1}^4 (b_{r_i} + w_{r_i}) - (1-d)$. For any particular cell, the quantity $\sum_{i=1}^4 y_i - d = \sum_{i=1}^4 (b_{r_i} + w_{r_i}) - 1$ is a constant. The value of y_i may be thought of as the distance, in coordinate i, between the segment and the upper boundary of the cell. If we sum these distances over all 4 coordinates, and then subtract the length of the segment, we obtain some fixed value. Because $b_{r_i} \le s_i \le b_{r_i} + w_{r_i}$, the distances must be in the range $0 \le y_i \le w_{r_i}$. In order for the segment to lie within the cell, it must be the case that $s_1 + d \le b_{r_1} + w_{r_1}$ and $s_2 + d \le b_{r_1} + w_{r_1}$. As $y_i = b_{r_i} + w_{r_i} - s_i$, these inequalities imply that $d \le y_1$ and $d \le y_2$.

To obtain an upper bound on the density of the cutting scheme on any segment in each cell, we solve a constrained optimization problem. For a particular cell (r_1, r_2, r_3, r_4), we maximize $\delta(r_1, r_2, r_3, r_4)$ over the variables y_i and d, subject to the constraints $\sum_{i=1}^4 y_i - d = \sum_{i=1}^4 (b_{r_i} + w_{r_i}) - 1$, $0 \le y_i \le w_{r_i}$, and $0 < d \le y_1, y_2$. We may perform the optimization using Lagrange multipliers. Alternatively, because all the density bounds are functions of terms that are only quadratic in the variables, we may use quadratic programming techniques to maximize the bounds. The resulting upper bounds that we obtain for the different cells are as follows.

$$\delta(1,1,1,1) \leq \frac{18765739}{16000000}$$
$$\delta(1,1,1,2) \leq \frac{21451739}{19200000}$$
$$\delta(1,1,1,3) \leq \frac{26890889}{24000000}$$
$$\delta(1,1,2,2) \leq \frac{32195739}{32000000}$$
$$\delta(1,2,1,1) \leq \frac{56447633}{48000000}$$
$$\delta(1,2,1,2) \leq \frac{106149377}{96000000}$$
$$\delta(2,2,1,1) \leq \frac{28177483}{24000000}$$
$$\delta(1,3,1,1) \leq \frac{56455889}{48000000}$$

Of these upper bounds, the largest is 56455889/48000000. This shows that the maximum density of the cutting scheme is at most $56455889/48000000 < 1.1762$. Therefore, when this cutting scheme is used as the rounding scheme in the Călinescu et al. algorithm, the approximation ratio will be at most 1.1762.

Chapter 6

Conclusion

In this project, we have studied rounding algorithms for the case of $k = 4$ terminals of the geometric relaxation given by Călinescu, Karloff, and Rabani for the multiway cut problem. We have defined several types of cuts of the 4-simplex, which we refer to as pair-isolating cuts, that were not previously studied by Călinescu et al. or by Karger, Klein, Stein, Thorup, and Young. Our interest in pair-isolating cuts is based on the assumption that the symmetry that they exhibit with respect to pairs of vertices makes them candidates for inclusion in cutting schemes that achieve small approximation ratios when used in the Călinescu et al. approximation algorithm for the multiway cut problem.

We have analyzed several natural cutting schemes in which a pair-isolating cut is chosen uniformly at random from a set of possible cuts. Our analysis suggests that pair-isolating cuts are not as effective as side-parallel cuts when used in the rounding scheme for the Călinescu et al. algorithm. This conclusion gains additional support from the results of our computational experiments, in which we have found cutting schemes involving sparcs and pair-isolating cuts by formulating the problem of producing an optimal cutting scheme as a discrete linear program. Based on our experiments, we may postulate that when only one of the different types of cuts is used in a cutting scheme, sparcs will yield a smaller approximation ratio than pair-isolating cuts.

When included in a cutting scheme in combination with sparcs, however, pair-isolating cuts appear to be useful. For each of the discrete primal linear programs involving sparcs and pair-isolating cuts that we solved, the optimal solution assigned positive probability to some pair-isolating cuts. For a fixed grid size N, the approximation ratio of an optimal solution to a primal program including both sparcs and pair-isolating cuts was smaller than the corresponding ratio for a program including only sparcs. While the discrete nature of the linear programs could distort the optimal solutions, these results provide evidence that, unlike in the case of $k = 3$, an optimal cutting scheme for $k = 4$ may not be specified purely as a probability distribution over sparcs.

Finally, we have given a sparc cutting scheme for $k = 4$ for which we may guarantee that it achieves a certain approximation ratio through an analytic proof. In this cutting scheme, there is a dependence among the probability distributions of the different slices that comprise the cut. The approximation ratio of this cutting scheme improves upon the smallest previously-known approximation ratio proven analytically.

6.1 Future Work

There are several natural directions in which to pursue the further study of the Călinescu et al. geometric relaxation for multiway cut. First, the question of the integrality gap for $k = 4$ remains open. Limited computational resources prevented us from solving the discrete linear programs involving pair-isolating cuts for most grid sizes larger than 30, and we did not observe a convergence in the objective values of the optimal solutions, as Karger et al. did for sparcs in the case of $k = 3$.

Despite a number of attempts, we were unable to find a cutting scheme using only pair-isolating cuts for which we could prove a maximum density less than 5/4. For sparcs, the work of Călinescu et al. and Karger et al. shows that there are relatively simple cutting schemes with maximum densities less than or equal to 5/4. It is possible that by proving that some pair-isolating cutting scheme has a small maximum density, one could gain some insight into the behavior of pair-isolating cuts that would allow for the development of better cutting schemes.

The results of our computational experiments suggest that sparcs alone are not sufficient for the construction of an optimal cutting scheme for $k = 4$. As such, introducing other k-way cuts as alternatives to sparcs for $k > 4$ is also a promising direction that one could pursue. When a side-parallel hyperplane is used to partition the simplex, the subset in which a particular point falls is determined by the value of a single coordinate of the point. In this work, we have defined hyperplanes for which a combination of two of the coordinate values of a point determine where the point lies in relation to the hyperplane. For pair-isolating slices, the relevant quantity is the sum of two coordinate values of a point, while for edge-perpendicular slices, the quantity of importance is the difference between two coordinate values. In the case of $k = 4$, three coordinate values of a point in the simplex determine the fourth coordinate value, and so it is unlikely that it would be useful to use hyperplanes for which more than two coordinate values would determine where a point lies in relation to the hyperplane. When $k > 4$, however, such hyperplanes could be helpful in furthering our understanding of the geometric relaxation.

Bibliography

[1] G. Călinescu, H. Karloff, and Y. Rabani. An improved approximation algorithm for Multiway Cut. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 48-52, Dallas, Texas, 23-26 May 1998.

[2] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23 (4): 864-894, August 1994.

[3] A. Freund and H. Karloff. A lower bound of $8/(7+\frac{1}{k-1})$ on the integrality ratio of the Călinescu-Karloff-Rabani relaxation for multiway cut. *Information Processing Letters*, 75 (1-2): 43-50, 31 July 2000.

[4] D. R. Karger, P. Klein, C. Stein, M. Thorup, and N. E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 668-678, Atlanta, Georgia, 1-4 May 1999.

[5] H. S. Stone. Multiprocessor scheduling with the aid of network flow algorithms. *IEEE Transactions on Software Engineering*, 3 (1): 85-93, January 1977.