Routing, Disjoint Paths, and Classification

SHUHENG ZHOU

August 2006

CMU–PDL–06–109

Dept. of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Thesis committee

Professor Avrim Blum (Carnegie Mellon University)
Professor Gregory R. Ganger, Co-chair (Carnegie Mellon University)
Professor Bruce M. Maggs, Co-chair (Carnegie Mellon University)
Professor Satish Rao (University of California, Berkeley)

© 2006 Shuheng Zhou. All Rights Reserved

To my mom and dad who give me my life and meaning
To my grandparents who nurtured me as a child
To Jun who gives himself to me

Abstract

In this thesis, we study two classes of problems: routing and classification. Routing problems include those that concern the tradeoff between routing table size and short-path forwarding (Part I), and the classic Edge Disjoint Paths problem (Part II). Both have applications in communication networks, especially in overlay network, and in large and high-speed networks, such as optical networks. The third part of this thesis concerns a type of classification problem that is motivated by a computational biology problem, where it is desirable that a small amount of genotype data from each individual is sufficient to classify individuals according to their populations of origin.

In hierarchical routing, we obtain "near-optimal" routing table size and path stretch through a randomized hierarchical decomposition scheme in the metric space induced by a graph. We say that a metric (X,d) has *doubling dimension* $\dim(X)$ at most α if every set of diameter D can be covered by 2^{α} sets of diameter $D/2$. (A *doubling metric* is one whose doubling dimension $\dim(X)$ is a constant.) For a connected graph G, whose shortest path distances d_G induce the doubling metric (X,d_G), we show how to perform $(1+\tau)$-stretch routing on G for any $0<\tau\leq 1$ with routing tables of size at most $(\alpha/\tau)^{O(\alpha)}\log\Delta\log\delta$ bits with only $(\alpha/\tau)^{O(\alpha)}\log\Delta$ entries, where Δ is the diameter of G and δ is the maximum degree of G. Hence, the number of routing table entries is just $\tau^{-O(1)}\log\Delta$ for doubling metrics.

The Edge Disjoint Paths (EDP) problem in undirected graphs refers to the following: Given a graph G with n nodes and a set $\mathcal{T}$ of pairs of terminals, connect as many terminal pairs as possible using paths that are mutually edge disjoint. This leads to a variety of classic NP-complete problems, for which approximabil-

ity is not well understood. We show a polylogarithmic approximation algorithm for the undirected EDP problem in general graphs with a moderate restriction on graph connectivity: we require the global minimum cut of G to be $\Omega(\log^5 n)$. Previously, constant or polylogarithmic approximation algorithms were known for trees with parallel edges, expanders, grids and grid-like graphs, and, most recently, even-degree planar graphs. These graphs either have special structure (e.g., they exclude minors) or there are large numbers of short disjoint paths. Our algorithm extends previous techniques in that it applies to graphs with high diameters and asymptotically large minors.

In the classification problem, we are given a set of $2N$ diploid individuals from population P_1 and P_2 (with no admixture), and a small amount of multilocus genotype data from the same set of K loci for all $2N$ individuals, and we aim to partition P_1 and P_2 perfectly. Each population P_a, where $a \in \{1,2\}$, is characterized by a set of allele frequencies at each locus. In our model, given the population of origin of each individual, the genotypes are assumed to be generated by drawing alleles independently at random across the K loci, each from its own distribution. For example, each SNP (or Single Nucleotide Polymorphism) has two alleles, which we denote with bit 1 and bit 0 respectively. In addition, each locus contains two bits (one from each parent) that are assumed to be two random draws from the same Bernoulli distribution.

We use p_1^k and $p_2^k, \forall k = 1,\ldots,K$ to denote frequency of an allele mapping to bit 1 at locus k in P_1 and P_2, respectively. We use $\gamma = \frac{\sum_{i=1}^{K}(p_1^i - p_2^i)^2}{K}$ as the dissimilarity measure between P_1 and P_2. We compute the number of loci K that we need to perform different tasks, versus N and γ, and prove several theorems. Ultimately, we show that with probability $1 - 1/\operatorname{poly}(N)$, given that $K = \Omega(\frac{\log N \log\log N}{N\gamma^2})$ and $K = \Omega(\frac{\log N}{\gamma})$, we can *recognize* the perfect partition (P_1, P_2) from among all other balanced partitions of the $2N$ individuals. We proved this theorem for two cases: either we are given two random draws for each attribute along each dimension, or only one.

Acknowledgements

I have been looking forward to writing this part for many years. It has been an odyssey, but by the end of this journey, temporarily I am unlost at the moment.

I thank my advisor Greg Ganger, without whom I could not have finished graduate school. At every darkest moment, Greg has always showed me his inspiring optimism and his unwaivering faith in me. He has given me full freedom to make some rather astonishing decisions, such as taking a leave of absence at the start of my third year and changing to theory at the end of my fourth year, while at the same time, carefully pushes me just a tiny bit, at the moment that I needed it, to overcome my self-doubts and to move beyond one more hurdle. I always end up feeling tremendously encouraged and rejuvenated after talking to him, because he always keeps me focused on the important issues given any situation, while offering the most practical and effective advice on how to get things done.

Greg has supported me selflessly, beyond what I could ever imagine or hope, throughout my entire graduate school years, while at the same time, he has always encouraged me to collaborate with other people. He remains my central source of support, both financially and morally, even after I switched to do theory. I thank Greg eternally for his genuine concern and his ultimate generosity toward me. Over the years, Greg has influenced me profoundly and shaped my view of life as a tireless researcher, through his rigorous work ethic and his consistently positive attitude toward extremely frustrating situations. His spirit, vision, patience, and passion toward research has always been the most inspiring. I could only wish that through incessant hard work, I can justify a tiny bit of his kindness and his trust in me. I also thank Greg profoundly for insisting that I think through the idea of embedding locality information in node identifiers in Distributed Hash Tables, which gave me the urge to understand deeply about hierarchical routing, and led to the spark of recognition for pursuing a theory path (and I have not drowned so far).

I thank my advisor Bruce Maggs for leading me through the door to theory. He not only advised me through the first part of my thesis work, but also got me interested in congestion related problems. Bruce has been extremely patient, supportive and fun to work with. Although he only has the last minutes to do things, he does them superbly quickly and accurately. And I appreciate him for being a good friend and for teaching me how to communicate more effectively in English. I also thank Bruce for supporting my traveling in the last year of graduate school.

I would like to express my gratitude to Satish Rao. During the last year of my visit at Berkeley, I had the rare privilege of working with Satish. The hours I spent discussing with Satish were always enlightening, and sustained me to spend every quiet morning and afternoon unraveling ideas that Satish proposed the day before or during the day. That was also the only period of my graduate school life that I actually spent the beautiful mornings (with the stunning California sunshine) working, instead of sleeping. I gained significant insights into these topics and a tremendous amount of courage in tackling difficult theory problems and in research in general; so I attribute this intensive period of intellectual growth and critical thinking to the guidance and nurture of Satish entirely.

I would like to thank my thesis committee. Avrim Blum has been most encouraging and responsive about my various queries in the last 10 months. I appreciate his feedbacks on the problems that I solve, and his suggestions on relevant questions and new directions to explore; I still owe him answers for most of these questions. I have always admired his broad scope of work in theory, but I could never have expected encouragement from him for tackling problems in new areas that I became interested in, such as learning theory.

I would like to thank the Berkeley theory group for allowing a stranger to share their lunches during my last year of graduate school; in particular, I thank the generous support from Satish for providing me an office space in Soda Hall and a computer account. It has been a pleasant and productive year of research and I am indebted to Berkeley for my productivity. I thank Professor Christos Papadimitriou and Professor Richard Karp for being interested in my research, and Henry Lin for discussing Game Theory and problems in routing with me.

I thank people from the CMU theory group for their support and interest, especially Harald Räcke for working with me on two difficult problems and Oleg Pikhurko for many delightful discussions. I thank Manuel Blum for the articles he put outside his office, some of which actually beautifully composed by himself, which I got to read on some late nights on my way to the CS lounge; from his article, I learned to write down what I read, which has been the only important thing that I know about how to do theory. I thank Manuel for being an inspiration. His eternal smile contains the warmth that melts away ice on the heart of any young graduate student or researcher, and I am not the first one to say this. I thank Hubert Chan and Nina Balcan for being most pleasant and candid.

At CMU, I also would like to thank Peter Steenkiste for being a nurturing advisor to Jun and being our constant support over the years; my first piece of research after I came back from industry was done under Peter's guidance – I appreciate his constant patience for listening to my vaguest ideas. I thank Mike Reiter for teaching the great introductory security class and for his concern about my progress and general happiness in graduate school. I thank Hui Zhang for giving me valuable advice that have helped me to evaluate the distance between dreams and reality. I thank Mor Harchol-Balter and John Lafferty for being encouraging and being available in answering many technical questions. I thank John especially for always listening to me encouragingly and patiently when I was scribbling on his blackboard. I thank Srini Seshan for discussing research ideas for one semester.

The Parallel Data Lab has been a warm home and an ideal place to work in. Karen Lindenfelser, Joan Digney, and Linda Whipkey are my strongest supporters and sweetest friends. The PDL students have been an exceptional group to be with; I thank all of them, especially those that I have been together with since the very beginning: Garth Goodson, my cubicle-mate for the longest time, John Griffin, Andy Klosterman, Chris Lumb, Jiri Schindler, Steve Schlosser, Craig Soules, and John Strunk. I thank the younger generation of PDL students that I had a chance to know: Michael Abd-El-Malek, James Hendricks, Michael Mesnier, Adam Pennington, Brandon Salmon, Minglong Shao, and Eno Thereska. I want to give special thanks to Jay Wylie, who has been my editor friend since I came back to PDL in Fall 2002. I thank people from the security group for their friendship, Lujo Bauer, Alina and Florin Oprea, Asad Samar, Dawn Song, Anat Talmy, and Chenxi Wang.

I thank the warm and cozy environment of ECE. Elaine Lawrence, my favorite lady, always holds an endless reservoir of patience and tolerance toward my various procrastination, and Lynn Philibin has kept close track of my progress. I thank all my dancers and performers, Claire Fang, Ginger Perng, Chris Lumb, Jennifer Morris, James Newsome, Craig Soules, Eno Thereska and Yang Wang, for their willingness to try and to appear in my one and only stage production for the ECE winter party 2003. I owe Yan Ke for finding me the music that made our production a success. You all enriched my life and my fondness of graduate school days.

I would like to thank many people that have influenced me at different stages of my academic pursuit and my choices; in particular, I mention at least a few. Professor Shi-Kang Guo from Tsinghua University has been a guarding angel for my soul since my last two years in college. I would like to thank Professor Deborah Estrin for the fascinating Networking class that I took at USC with her; it is her who taught me the first step of doing research and understanding the basics of a field by reading a lot of papers. I would like to thank Steven McCanne and Scott Shenker for initiating and arranging my last year's visit at Berkeley; it is nice being welcomed, and I am grateful to Scott for granting me an extended invitation for using his desk at ICSI for the years to come. I appreciate their concern and their generosity forever. I want to thank Steve for his beautifully-written thesis, and the inspiring work behind it; it was the first evidence to me that doing research is an attractive and fascinating task, and the labor behind it is totally worth it.

I would like to thank Lisa Zhang and Matthew Andrews for their discussions with me about various hardness results in their papers. I especially thank Lisa for her friendship and her extensive help during my job search stage. I thank Bobby Kleinberg for providing me invaluable advice on many things, and for his enthusiasm and his kindness in general. I thank Jon Kleinberg for his interest in my work.

I would like to thank my friends at Pittsburgh over the years, Ping Gao, Ying Shi, Jiantao Pan, Yan Lu and Peng Chang, Xuemei Gu and Kan Deng, Yinglian Xie and Qifa Ke, Yanghua Chu and Wendy Chen, Wenrui Mi and Haotian Zhang, Grace and Thomas Nordin, Fang Fang and Yan Ke, and for making the summer days more lively and the winter nights cozier. I also thank Julio Lopez, Desney Tan, Umut Acar and Hal Burch, who have always been the most pleasant people to encounter on campus. I admire each of them for their individual passions. I thank Yaping Li and Jiyang Li for the wonderful time that we spent at Berkeley, and Limin Jia at Princeton.

I would like to thank my friends from my Southern California days, with whom I have shared my memorable younger days in America, and somehow every day of that year was an exciting day in my life – Lan Guo and Rundong Huang, Cissy Fang Liu, Miao Fu, Bang Du, Ping Jiang and Gang Yu. I thank them for sharing my wildest dreams, caring about me, and being tolerant of my insensitivity (with an excuse of being young at that time), and insisting on buying me numerous dinners that I have not had a chance return. I especially appreciate Lan for her continuous concern about my progress both in life and in school after I move to CMU. I thank my college friends and the girls from my dorm, for their friendship and their love when I was very young. Especially, I thank Yingying Liu, Jinjiang Li, Yifan Li, Hui Wan, Dong Wang, Donghui Yu, and Alex Zhu for keeping in touch.

Finally, I treasure the little children who are growing up during my graduate school years, for their inspiration to me as someone who always longs for innocence and creativity: Emily Cai, David Chang, and Emily Ke. The most important lesson that I learned from them is: happiness is a dry diaper.

I would like to thank our fondest American family: the Brenizers, Diana, Jack, Jon and Joan. They are most loving and caring toward me and Jun. I am forever indebted to Diana for arranging everything for our wedding at State College, and I feel tremendously blessed for knowing them. I thank Jun, who shared most of my frustrations and struggles throughout my graduate school years. He kept me focused on my research, especially during the time I work on theory, which has demanded more attention from me than anything else in life. He always takes me to places that I like to be, shows genuine interests in things that I do (even Ballet), enjoys the type of food that I like, and never blames me for anything – he is my sunshine and my happiness. I thank Jun for providing the warmest and most peaceful place in the world – wherever we are together.

I thank my parents, for being totally supportive, never doubtful, and always encouraging and positive. I thank them for providing me the best education and the best quality of life, even with their moderate income. Most of all, I thank them for loving me and caring about me unceasingly. I want to thank my grandparents, for taking care of me when I was a child. I thank my grandfather for being my first teacher in mathematics. I cherish my extended family and Jun's family for their love and their acceptance of me.

I cherish the beautiful music and songs of those great artists that have always touched my heart deeply and sustained me through saddest moments of my life: Mozart, Beatles, and Roxette. I would like to thank a very special friend of mine, Daria Ward, for not only teaching me how to dance, but for showing me how to be kind and open to the whole world. This is for the city of Pittsburgh. I thank the generosity, the friendliness and the warmth from the people of Pittsburgh.

Finally, I thank my hometown and its vastly open and unpolluted frontier land – at least 20 years ago – in the most northeastern part of China. It has the harsh, long, but extremely clean winter that I loved most as a child and teenager. It taught me the things that are most precious to me: dream, courage, endurance, devotion, faith, hope, innocence, purity, openness, and sincerity. Those have not changed since I was 16, when I left home.

This thesis is finished at a time when the world is in trouble, and the health of my dearest grandmother is in need of prayers; I thank all those people, especially my mom and my uncle, who make the relative peace of mind in this space, at this moment possible, so that I can have the luxury to finish my writing. Thank you from the bottom of my heart.

I thank my co-authors: Anupam Gupta, Eran Halperin, Bruce Maggs, and Satish Rao for their contributions to this thesis; Chapter 2 is joint work with Anupam Gupta and Bruce Maggs. Chapter 4-6 is joint work with Satish Rao. Chapter 7 and 8 are based on a computational biology project with Eran Halperin and Satish Rao. Proofs in Chapter 8 are joint work with Satish Rao. The idea for an alternative score in Chapter 9 comes from Avrim Blum.

I thank the members and companies of the PDL Consortium (including APC, EMC, Engenio, Equallogic, Hewlett-Packard, HGST, Hitachi, IBM, Intel, Microsoft, Network Appliance, Oracle, Panasas, Seagate, Sun, and Veritas) for their interest, insight, feedback, and support. This material is based on research sponsored in part by the Army Research Office, under agreement number DAAD19–02–1–0389, and NSF grant CNF–0435382.

Contents

Figures

Tables

1 Introduction

This thesis concerns three problems: hierarchical routing, edge-disjoint paths, and classification.

1.1 Hierarchical Routing and Hierarchical Decompositions

In seminal work by Kleinrock and Kamoun [1977], a hierarchical routing scheme based on an "optimal" hierarchical clustering model of nodes in the network is described. They further show that for a class of large distributed networks, by following their routing scheme, it is possible to achieve a substantial reduction in routing table size with essentially no increase in the *average path length*, over all source-destination pairs in the network.

Essentially, the family of networks upon which it is possible to apply such an "optimal" hierarchical clustering scheme satisfies certain growth properties such that: (a) the diameter of any cluster S of nodes chosen is bounded above by $O(|S|^{\nu})$ for some constant $\nu \in [0,1]$, and (b) the average distance between nodes in the network is $\Theta(N^{\nu})$, where N is the size of the network.

While some recent papers by Plaxton et al. [1999]; Karger and Ruhl [2002]; Hildrum et al. [2002] on distributed object location in peer-to-peer networks used definitions and restrictions that differ slightly from each other, the essential theme was to reduce the "intrinsic complexity" of each problem in its own context by bounding the growth rate of networks, as done by Kleinrock and Kamoun.

We design the piece that is missing from Kleinrock and Kamoun [1977]: a hierarchical decomposition algorithm. We further improve their results by giving bounds on path stretch on a *per node-pair* level using slightly different assumptions on the network growth. Specifically, we capture the network growth and parameterize the inherent “complexity” of a metric space (X,d) generated by such a network using its *doubling dimension* $\dim(X)$: the least value α such that each ball of radius R can be covered by at most 2^{α} balls of radius $R/2$.

We show the following result.

Theorem 1.1. *Given any network G, whose shortest path distances d_G induce the doubling metric (X,d_G) with $\dim(X) = \alpha$, and any $\tau > 0$, there is a routing scheme on G that achieves $(1+\tau)$-stretch, where each node stores only $(\frac{\alpha}{\tau})^{O(\alpha)} \log\Delta \log\delta$ bits of routing information, where Δ is the diameter of G and δ is the maximum degree of G.*

Note that for any $\alpha \in \mathbb{Z}$, the space $\mathbb{R}^{\alpha}$ under any of the ℓ_p norms has doubling dimension $\Theta(\alpha)$, and hence this doubling dimension extends the standard notion of geometric dimension. This also allows us to conclude that, in order to obtain a near-optimal routing scheme in terms of path stretch and routing table size, all we need is a simple restriction on how fast the network grows.

1.2 Edge-Disjoint Paths in Moderately Connected Graphs

In the second part of this thesis, we first explore approximation for the edge disjoint paths (EDP) problem: Given a graph with n nodes and a set of terminal pairs, connect as many of the specified pairs as possible using paths that are mutually edge disjoint. EDP has a multitude of applications in areas such as VLSI design, routing and admission control in large-scale, high-speed and optical networks. Moreover, EDP and its variants have also been prominent topics in combinatorics and theoretical computer science for decades. For example, the celebrated theory of graph minors by Robertson and Seymour [1990] gives a polynomial time algorithm for routing all the pairs given a constant number of pairs. However, varying the number of terminal pairs leads to a variety of classic NP-complete problems, for which approximability is an interesting problem. In a recent breakthrough, Andrews and Zhang [2005b] showed an $\Omega(\log^{\frac{1}{3}-\varepsilon} n)$ lower bound on the hardness of approximation for undirected EDP.

In this work, we show a polylogarithmic approximation algorithm for the undirected EDP problem in general graphs with a moderate restriction on graph connectivity: we require that there are $\Omega(\log^5 n)$ edge disjoint paths between every pair of vertices, i.e., the global min cut is of size $\Omega(\log^5 n)$. If this moderately connected case holds, we can route $\Omega(\mathsf{OPT}/\operatorname{polylog} n)$ pairs using disjoint paths with congestion 1, where OPT is the maximum number of pairs that one can route edge disjointly for the given EDP instance. Previously, constant or polylogarithmic approximation algorithms were known for trees with parallel edges, expanders, grids and grid-like graphs, and, most recently, even-degree planar graphs by Kleinberg [2005]. The results rely either on excluding a minor (or other structural properties) or the fact that many very short paths exist. Our algorithm extends previous techniques; for example, our graphs can have high diameter and contain very large minors. We are hopeful that this constraint on the global minimum cut can be removed if congestion on each edge is allowed to be $O(\log\log n)$. Formally, we have the following result.

Theorem 1.2. *There is a* $\operatorname{polylog} n$*-approximation algorithm for the edge disjoint paths problem in a general graph* $\mathcal{G}$ *with minimum cut and node degree* $\Omega(\log^5 n)$.

1.3 Population Classification

In the third part of this thesis, we explore a type of classification problems in the context of a computational biology problem. In particular, we aim to classify individuals according to their populations of origin, based on only a small amount of their genotype data.

In seminal work by Pritchard et al. [2000], two types of clustering methods are described for using multilocus genotype data to infer population structure and assign individuals to populations.

(1) *Distance-based Methods.* These proceed by calculating a pairwise distance matrix, whose entries give the distance between every pair of individuals. This matrix may then be represented using some convenient graphical representation (such as a tree or a multidimensional scaling plot) and clusters may be identified by eye.

(2) *Model-based Methods.* These proceed by assuming that observations from each cluster are random draws from some parametric model. Inference for

> the parameters corresponding to each cluster is done jointly with inference for the cluster membership of each individual, using standard statistical methods (for example, maximum-likelihood or Bayesian methods).

A model-based clustering method is used by Pritchard et al. [2000]. They assume a model in which there are K populations, where K may be unknown, and each population is characterized by a set of allele frequencies at each locus.

While we follow essentially the same model, assuming no admixture, we fix $K = 2$. We name our method *graph-based*; in some sense, it is similar to *Distance-based Methods* in that we assign a score to every pair of individuals that capture the degree of dissimilarity between them; the true novelty of our approach, however, is that we construct a complete graph while assigning scores to edges, such that in expectation, a balanced cut with the maximum score, which we denote as the *max-cut* of the complete graph, will provide us the perfect partition – i.e., the perfect partition indeed has the maximum score among all balanced partitions in the complete graph, given a balanced input instance.

Our goal is to minimize the number of loci that we require in order to classify the two populations, given a set of $2N$ diploid individuals from two populations P_1 and P_2 and their genotypes from the same set of K loci. Recall that for diploid organisms the chromosomes come in pairs. A genotype is a list of unordered pairs of alleles, such that one comes from each of the parents.

Since each Single Nucleotide Polymorphism (SNP) has two variants (alleles), we use bit 1 and bit 0 to denote them. Given the population of origin of each individual, the genotypes are assumed to be generated by drawing alleles independently from the appropriate population frequency distribution. We use p_1^k and p_2^k to denote the "success" probability (frequency of an allele mapping to bit 1) at locus k in the population of origin 1 and 2 respectively. Each locus contains two bits that are assumed to be two random draws from the same Bernoulli distribution. We use $\gamma = \frac{\sum_{k=1}^{K}(p_1^k - p_2^k)^2}{K}$ as the measure that we optimize the number of loci we need against. We show three results whose proof ranges from straight-forward to sophisticated.

1.3.1 Quartet-based Scores

In all three theorems, we assign the same score to a pair of individuals X,Y, which measures the difference between the two individuals, hence a higher score is more desirable for two individuals from different populations of origin. Using this score, we can construct a complete graph where nodes are individuals and edge weight is the score between the two individuals. Note that when we say the score for a cut, we mean the sum of scores on all edges across the cut. In particular, we call this score $\mathsf{Pscore}(X,Y)$ for an unordered pair of individuals (X,Y):

Definition 1.1.

$$\mathit{Pscore}(X,Y) = \sum_{i=1}^{K} \mathit{Pscore}^i(X,Y) = \sum_{i=1}^{K} \mathit{Pscore}^i \begin{bmatrix} x_1^i & x_2^i \\ y_1^i & y_2^i \end{bmatrix},$$

where

$$\mathit{Pscore}^i(X,Y) = \frac{1}{2} \left(\begin{array}{c} (I_{x_1^i=x_2^i} + I_{y_1^i=y_2^i}) - (I_{x_1^i=y_1^i} + I_{x_2^i=y_2^i}) + \\ (I_{x_1^i=x_2^i} + I_{y_1^i=y_2^i}) - (I_{x_1^i=y_2^i} + I_{x_2^i=y_1^i}) \end{array} \right),$$

where $I_{x=y} = 1$ if $x = y$, and $= 0$ otherwise.

Note that this definition utilizes an important quartet construction involving four bits $x_1^i, x_2^i, y_1^i, y_2^i$, which are four independent Bernoulli random variables, such that two bits from each pair (x_1^i, x_2^i), (y_1^i, y_2^i) are identically distributed.

The first theorem says that, given enough loci, all scores are *correct* in the following sense.

Theorem 1.3. (Global Optimum Lemma) *Let* $2N$ *be the total number of individuals. Given that* $K \geq 18 \ln N/\gamma^2$*, with probability* $1 - O(1/N^2)$*, for all quartets* X,Y,Z_1,Z_2 *such that* X,Y *come from different populations, while* Z_1,Z_2 *come from the same population,*

$$\mathit{Pscore}(X,Y) \geq \mathit{Pscore}(Z_1,Z_2).$$

This immediately implies that, given a balanced input instance where we have the same number of individuals from each population, the *max-cut* (i.e., the cut with the maximum score) separates the two populations perfectly. This gives us the trivial algorithm for separating a balanced input instance into P_1 and P_2 and assigning individuals correctly: simply keep the top N^2 edges in terms of Pscore in the complete graph, and these edges correspond to a *max-cut* that separates P_1 from P_2 perfectly.

For imbalanced input instances, the perfect partition (P_1, P_2) has the maximum *average* score, where *average* score for a cut is defined as the total score across edges in the cut divided by the number of such edges.

In addition, for imbalanced instance, we only need to adjust the constant in the bound for K, so that all edges between individuals from different populations are above a certain threshold h while all other edges are blow threshold $\ell < h$, given that the expected values for $\mathsf{Pscore}(X,Y)$ and $\mathsf{Pscore}(Z_1,Z_2)$ differ significantly from one another: $\mathbf{E}[\mathsf{Pscore}(X,Y)] \geq 2K\gamma$ while $\mathbf{E}[\mathsf{Pscore}(Z_1,Z_2)] = 0$. By keeping only edges above a certain threshold and by taking account of deviation, we keep edges that define a perfect partition. Hence, this algorithm works for both input cases.

We call this theorem a *Global Optimum Lemma*, since there exists an overall desirable ordering among all edge scores in the complete graph.

The second theorem says that, given we have some pre-classified individuals from P_1 and P_2, N from each origin, it requires fewer bits from a new individual in order to put it on the correct side, since the sum of dissimilarity scores from this new individual X to the other population is consistently higher than the sum of scores to its own population.

Theorem 1.4. (Local Optimum Lemma). *Let* $K \geq \max\{\frac{9\ln(1/\delta)}{N\gamma^2}, \frac{8\ln(1/\tau)}{\gamma}\}$. *For any X, w.l.o.g. from* P_1, *and its observed bit string* $\tilde{X}$, *with probability* $1-\tau-\delta$, *given that* $X_i, Y_i, \forall i$ *are individuals randomly draw from* P_1 *and* P_2 *respectively, we have*

$$\sum_{i=1}^{N} \mathit{Pscore}(X, Y_i | X = \tilde{X}) > \sum_{i=1}^{N} \mathit{Pscore}(X, X_i | X = \tilde{X}).$$

A similar statement holds for any Y from P_2.

This tells us that once we have $2N$ individuals, N from each population, that are already classified properly, a new node can almost always pick the correct side to join based on its local view: it just needs to join the side that it has a lower total score to the N individuals on that side. Hence, we denote this theorem as a *Local Optimum Lemma*.

Finally, we show a theorem says that perfect partition corresponds to the *max-cut* in the complete graph, given any balanced input instance, by requiring slightly more bits than necessary in the Local Optima Lemma, but still asymptotically fewer (in terms of γ) than that of Global Optima Lemma.

Theorem 1.5. *Given that $K = \Omega(\frac{\ln N}{\gamma})$ and $KN = \Omega(\frac{\ln N \log\log N}{\gamma^2})$, where $N \geq 8$, with probability $1 - 1/\operatorname{poly}(N)$, we can differentiate the perfect partition from all other balanced partitions of individuals.*

1.3.2 Learning Mixtures of Product Distributions

After exploring the power of two random draws from any one dimensional distribution in the K dimensional distributions, we ponder at the possibility of achieving the same power of clustering using a single random draw from each dimension: given a small sample, i.e., N is small, can we learn the partition with a small amount of attributes, if for each attribute, we are given only a single bit from its Bernoulli distribution?

The answer is positive. We show the following theorem using an inner product based score, which we call Rscore.

Definition 1.2. $\mathit{Rscore}(X,Y) = <\vec{x},\vec{y}> = \sum_{i=1}^{K} x^i y^i$.

Theorem 1.6. *Given that $K = \Omega(\frac{\ln N}{\gamma})$ and $KN = \Omega(\frac{\ln N \log\log N}{\gamma^2})$, where $N \geq 4$, with probability $1 - 1/\operatorname{poly}(N)$, for all balanced cuts in the complete graph formed among $2N$ sample points, we can differentiate the perfect partition from all other balanced partitions of the sample by finding the min-cut.*

We note that Hamming distance based score will give similar claim, using max-cut. We also note that neither Rscore nor Hamming distance based score will give us claims similar to Global or Local Optima Lemmas as in Theorem 1.3 and 1.5. However, for the special case that we know whether $\forall i, p_1^i \geq p_2^i$ or vice versa, then a simple bit-wise score that is similar to what we define below suffices to prove a Global Optimum Lemma; in particular suppose that we know $\forall i, p_1^i \geq p_2^i$:

Definition 1.3. *For an unordered pair of individuals* (X,Y)*, let* $\mathsf{Bscore}^i(X,Y) = (x^i - y^i), \forall i$.

$$\mathsf{Bscore}(X,Y) = \sum_{i=1}^{K} \mathsf{Bscore}^i(X,Y).$$

We show that the absolute value of scores between points from the same distribution is consistently below those between points from different distributions in Theorem 9.2.

1.3.3 Related Work

There are two streams of related work. The first stream is the recent progress in learning from the point of view of clustering, where given samples drawn from a mixture of well-separated Gaussians (component distributions), one aims to classify the sample according to which component distribution it comes from, as studied in Dasgupta [1999]; Dasgupta and Schulman [2000]; Arora and Kannan [2001]; Vempala and Wang [2002]; Achlioptas and McSherry [2005]; Kannan et al. [2005]; Dasgupta et al. [2005]. Under this framework, it has also been extended to more general distributions such as log-concave distributions in Achlioptas and McSherry [2005]; Kannan et al. [2005] and heavy-tail distributions in Dasgupta et al. [2005].

These work mostly focus on reducing the requirement on the sufficient separation conditions between any two centers P_1 and P_2 in the mixture from dependence on K, the dimensions, to dependence only on the number of components in the mixture, in order to classify most of the sample correctly. In contrast, we focus on the case that although we only have a mixture of two product distributions, the sample size, i.e., number of individuals, is small; we prove that by acquiring enough number of attributes along the same set of dimensions from both distributions, with high probability, we can correctly classify every node in the sample.

The second stream of work is under the Probably Approximately Correct (PAC) framework, where given a sample generated from some target distribution Z, the goal is to output a distribution Z_1 that is close to Z according to Kullback-Leibler divergence: $KL(Z||Z_1)$, where Z is a mixture of product distributions over discrete domains or Gaussians (Kearns et al. [1994]; Freund and Mansour [1999]; Cryan [1999]; Cryan et al. [2002]; Mossel and Roch [2005]; Feldman et al. [2005, 2006]). These work do not require a minimal distance between any two distributions.

To compare our results with learning mixtures of Gaussians, we first denote the ℓ_2-square distance between the centers of the two distributions: $||P_1 - P_2||_2^2 = K\gamma = \sum_{i=1}^{K}(p_1^k - p_2^k)^2$.

(1) Theorem 1.3 requires that the distance between two distributions: $||P_1 - P_2||_2 = \Omega^*(K^{1/4})$, i.e., the separation requirement depends on the number of dimensions of each product distribution. This is comparable to that in Dasgupta and Schulman [2000]; Arora and Kannan [2001].

(2) Theorem 1.5 requires that $d = ||P_1 - P_2||_2 = \Omega(\ln^{\frac{1}{2}} N)$, where $N = \Omega^*(K/d^4)$, which is independent of the dimension of the product distribution; this is comparable to what Kannan et al. [2005], and Achlioptas and McSherry [2005] accomplish for the continuous case.

1.4 Thesis Outline

Chapter 2 and 3 belong to Part I (Hierarchical Routing). Part II (Edge Disjoint Paths) contains Chapter 4–6. Chapters 7–9 belong to Part III (Classification). One can safely skip Chapter 5 while still being able to connect Chapter 4 with Chapter 6.

Part I: Hiearchical Routing

2 Hierarchical Routing in Doubling Metrics

2.1 Introduction

The *doubling dimension* of a metric space (X,d) is the least value α such that each ball of radius R can be covered by at most 2^{α} balls of radius $R/2$ Gupta et al. [2003]. For any $\alpha \in \mathbb{Z}$, the space $\mathbb{R}^{\alpha}$ under any of the ℓ_p norms has doubling dimension $\Theta(\alpha)$, and hence this doubling dimension extends the standard notion of geometric dimension; moreover, it can be seen as a way to parameterize the inherent “complexity” of metrics.

In this chapter, we study the problem of designing routing algorithms for networks whose structure is parameterized by the doubling dimension $\dim(X) = \alpha$; we show that one can route along paths with stretch $(1+\tau)$ with small routing tables—with only $O((\alpha/\tau)^{O(\alpha)} \log \Delta)$ *entries*, where Δ is the diameter of the network G. Each entry stores at most $O(\log \delta)$ bits, where δ is the maximum degree of G, and hence for doubling metrics—where α is a constant—and any $\tau \leq 1$, we have $(1+\tau)$-stretch routing with only $O(\log \Delta \log \delta)$ bits of routing information at each node.

The idea of placing restrictions on the growth rate of networks to bound their “intrinsic complexity” is by no means novel; it has been around for a long time (see, e.g., Kleinrock and Kamoun [1977]), and has recently been used in several contexts in the literature on object location in peer-to-peer networks Plaxton et al. [1999]; Karger and Ruhl [2002]; Hildrum et al. [2002]. While these papers used definitions and restrictions that differ slightly from each other, we note that our results hold in those models as well. Our results extend those of Talwar [2004], whose routing schemes for metrics with $\dim(X) = \alpha$ require local routing information of $\approx O(\log^{\alpha} \Delta)$ bits. Formally, we have the following main result.

Theorem 2.1. *Given any network G, whose shortest path distances d_G induce the doubling metric (X, d_G) with $\dim(X) = \alpha$, and any $\tau > 0$, there is a routing scheme on G that achieves $(1+\tau)$-stretch and where each node stores only $(\frac{\alpha}{\tau})^{O(\alpha)} \log \Delta \log \delta$ bits of routing information, where Δ is the diameter of G and δ is the maximum degree of G.*

The proof of the theorem proceeds along familiar lines; we construct a set of hierarchical decompositions (HDs) of the metric (X, d), where each HD consists of a set of successively finer partitions of X with geometrically decreasing diameters. Each node in X maintains a table containing next hops to a small subset of clusters in these partitions; to route a packet from s to t, we use the routing table for s to pick some "small cluster" C in s' table that contains t and send the packet to some node x in C; a similar process repeats at node $x \in C$ until the packet reaches t. The idea is to create routing tables which ensure that the distance from x to t is much smaller than that from s to t, and hence the detour taken in going from s to t is only $\tau d(s,t)$. (Details of routing schemes appear in Section 2.4 and 3.1.)

While this framework is well-known, the standard ways to construct HDs are top-down methods which iteratively refine partitions. These methods create long-range dependencies which require us to build $O(\log n)$ HDs in general; in order to use the locality of the doubling metrics and get away with $\tilde{O}(\alpha)$ HDs, we develop a bottom-up approach that avoids these dependencies when building HDs. The analysis of this process uses the Lovász Local Lemma (much as in Krauthgamer and Lee [2003]; Gupta et al. [2003]); details are given in Section 2.3.

2.1.1 Related Work

Distributed packet routing protocols have been widely studied in the theoretical computer science community; see, e.g., Frederickson and Janardan [1988, 1989]; Awerbuch and Peleg [1992]; Peleg and Upfal [1989]; Cowen [2001]; Peleg [2000], or the survey by Gavoille [2001] on some of the issues and techniques. Note that these results, however, are usually for general networks, or for networks with some topological structure. By placing restrictions on the doubling dimension, we are able to give results which degrade gracefully as the "complexity" of the metric increases. For example, it is known that any universal routing algorithm with stretch less than 3 requires *some* node to store at least $\Omega(n)$ routing information Gavoille and Gengler [2001]; however, these graphs generate metrics with large $\dim(X)$. Our results thus allow one to circumvent these lower bounds for metrics of "lower dimension".

Packet routing in low dimensional networks has been previously studied in Talwar [2004], that gives algorithms that require $O(\alpha(\frac{6}{\tau\alpha})^{\alpha}(\log^{\alpha+2}\Delta))$ bits of information to be stored per node in order to achieve $(1+\tau)$-stretch routing—for constant stretch τ and doubling dimension α. The resulting dependence of $O(\log^{2+\alpha}\Delta)$ should be contrasted with the dependence of $O(\log\Delta\log\delta)$ bits of information in our schemes. We should point out that his algorithms are based on graph decomposition ideas with a top-down approach and do not require the LLL to construct routing tables.

One of the papers that influence this work is that of Kleinrock and Kamoun [1977]. They describe a general hierarchical clustering model on which our routing schemes are based. They show that routing schemes based on a hierarchical clustering model do not cause much increase in the *average path length* for networks that satisfy the following two assumptions: (a) the diameter of any cluster S chosen is bounded above by $O(|S|^{\nu})$ for some constant $\nu \in [0,1]$, and (b) the average distance between nodes in the network is $\Theta(n^{\nu})$. In contrast, we give bounds on the path stretch on a *per node-pair* level using slightly different assumptions on the network geometry.

Other papers on object location in peer-to-peer networks Plaxton et al. [1999]; Karger and Ruhl [2002]; Hildrum et al. [2002] have also used restrictions similar to Kleinrock and Kamoun [1977] on the growth rate of metrics; in particular, they consider metrics where increasing the radius of any ball by a factor of 2 causes the number of points in it to increase by at most some constant factor 2^{β}. (Plaxton et al. Plaxton et al. [1999] also consider the *lower bound* on the growth.) Here the parameter β can be considered to be another notion of "dimension" for a metric space. It can be shown that $\dim(X) \leq 4\beta$ [Gupta et al., 2003, Prop. 1.2]; hence our results hold for such metrics as well. Our scheme is also similar in spirit to a data-tracking scheme of Rajaraman et al. [2001], who use approximations by tree distributions to obtain bounds on the stretch incurred.

2.2 Definitions and Notation

Let the input metric be (X,d); this paper deals with finite metrics with at least 2 points. We use standard terminology from the theory of metric spaces; many definitions can be found in Deza and Laurent [1997] and Heinonen [2001]. Given $x \in X$ and $r \geq 0$, we let $\mathbf{B}(x,r)$ denote $\{x' \in X \mid d(x,x') \leq r\}$, i.e., the ball of radius r around x. Given a subset $S \subseteq X$, the distance of $x \in X$ to the set S is $d(x,S) = \min\{d(x,x') \mid x' \in S\}$.

The *doubling constant* λ_X of a metric space (X,d) is the smallest value λ such that every ball in X can be covered by λ balls of half the radius. The *doubling dimension* of X is then defined as $\dim(X) = \log_2 \lambda_X$; we use the letter α to denote $\dim(X)$. A metric is called *doubling* when its doubling dimension is a constant. A subset $Y \subseteq X$ is an *r-net* of X if (1) for every $x,y \in Y, d(x,y) \geq r$ and (2) $X \subseteq \cup_{y \in Y} \mathbf{B}(y,r)$. Such nets always exist for any $r > 0$, and can be found using a greedy algorithm.

Proposition 2.1 (Gupta et al. [2003]). *If all pairwise distances in a set $Y \subseteq X$ are at least r (e.g., when Y is an r-net of X), then for any point $x \in X$ and radius t, we have that $|\mathbf{B}(x,t) \cap Y| \leq \lambda_X^{\lceil \log_2 \frac{2t}{r} \rceil}$.*

Proof. Applying the definition of *doubling constant* of the input metric (X,d), $\mathbf{B}(x,t)$ can be covered by λ balls of radius $t/2$ centered around some vertices inside $B(x,t)$. By applying the same definition at most $\lceil \log_2 \frac{2t}{r} \rceil$ times, one get a cover of $\mathbf{B}(x,t)$ with $\lambda^{\lceil \log_2 \frac{2t}{r} \rceil}$ balls of radius $\leq r/2$. Since all pairwise distances in $Y \subseteq X$ are at least r, none of $y,y' \in Y$ can fall into the same ball of radius $\leq r/2$; thus each ball of radius $r/2$ covers at most 1 node from Y. Thus we have $|\mathbf{B}(x,t) \cap Y| \leq \lambda^{\lceil \log_2 \frac{2t}{r} \rceil}$. □

A *cluster* C in the metric (X,d) is just a subset of points of the set X. The diameter of the cluster C is the largest distance between points of the cluster. Each cluster is associated with a *center* $x \in X$ (*which may not lie in* C) and the *radius* of the cluster C is the smallest value r such that the cluster C is contained in $\mathbf{B}(x,r)$.

Definition 2.1. *Given $r > 0$, an r-***ball partition** Π *of (X,d) is a partition of X into clusters $C_1, C_2, \ldots$, with each cluster C_i having a radius at most r.*

By scaling, let us assume that the smallest inter-point distance in X is exactly 1. Let Δ denote the diameter of the metric (X,d), and hence Δ is also the aspect ratio of the metric. Define $\rho = 256\alpha + 1$ and $h = \lceil \log_\rho \Delta \rceil$. Let us define $\eta_i = 1 + \rho + \rho^2 + \ldots + \rho^i < \rho^{i+1}/(\rho - 1)$; note that $\eta_i = \rho\,\eta_{i-1} + 1$. Let us fix a $\rho^i/2$-net and denote with N_i for the metric (X,d), for every $0 \le i \le h+1$.

2.2.1 Hierarchical Decompositions (HDs)

We now give a formal definition of a *hierarchical decomposition* (HD) which is used throughout this paper and is the basic object of our study. As noted below, such a decomposition can be naturally associated with a decomposition tree that is used for our hierarchical routing schemes.

Definition 2.2. *A* ρ-hierarchical decomposition $\mathbf{\Pi}$ *(*ρ-HD*) of the metric (X,d) is a sequence of partitions $\Pi_0, \ldots, \Pi_h$ with $h = \lceil \log_\rho \Delta \rceil$ such that:*

(*1*) *The partition Π_h has one cluster X, the entire set.*

(*2*) **(geometrically decreasing diameters)** *The partition Π_i is an η_i-ball partition. Since inter-point distances are at least 1, it implies that $\Pi_0 = \{\{x\} : x \in X\}$; in other words, each cluster in Π_0 is a singleton vertex.*

(*3*) **(hierarchical)** *Π_i is a refinement of Π_{i+1} and each cluster in Π_i is contained within some cluster of Π_{i+1}.*

Given such a ρ-HD $\mathbf{\Pi} = (\Pi_i)_{i=0}^{h}$, the partition Π_i is called the *level-i* partition of $\mathbf{\Pi}$ and clusters in Π_i are the *level-i* clusters. Note that these clusters have a radius η_i and hence diameter $\le 2\eta_i$. Furthermore, define the *degree* $\deg(\mathbf{\Pi})$ to be the maximum number of level-i clusters contained in any level-$(i+1)$ cluster in Π_{i+1}, for all $0 \le i \le h-1$.

Hierarchical Decompositions and HSTs. A hierarchical decomposition is a *laminar family* of sets, where given any two sets, they are either disjoint or one contains the other. It is well known that such a family $\mathcal{F}$ of sets over X can be associated with a natural decomposition tree whose vertices are sets in $\mathcal{F}$ and whose leaves are all the smallest sets in the family (which are elements of X, in this case). We can use this to associate a so-called hierarchically well-separated tree (also called an HST Bartal [1996]) $T_{\mathbf{\Pi}}$ with a hierarchical decomposition $\mathbf{\Pi}$; since each edge in $T_{\mathbf{\Pi}}$ connects some $C \in \Pi_i$ and $C' \in \Pi_{i-1}$ with $C' \subseteq C$, we associate a *length* η_i with edge (C, C'). Given such a tree $T_{\mathbf{\Pi}}$, we can (and indeed do) talk about its level-i clusters with no ambiguity; these are the same level-i clusters in the associated Π_i. Note that the degree of vertices in this tree $T_{\mathbf{\Pi}}$ is bounded by $\deg(\mathbf{\Pi}) + 1$.

2.2.2 Padded Probabilistic Ball-Partitions

Recall that an r-ball partition Π of (X, d) is a partition of X into a set of clusters $C \subseteq X$, each contained in a ball $\mathbf{B}(v, r)$ for some $v \in X$. $\mathbf{B}(x, t)$ is *cut* in the partition Π if there is no cluster $C \in \Pi$ such that $\mathbf{B}(x, t) \subseteq C$. In general, $\mathbf{B}(x, t)$ is *cut* by a set $S \subseteq X$ if both $S \cap \mathbf{B}(x, t)$ and $\mathbf{B}(x, t) \setminus S$ are non-empty.

Let $\mathcal{P}$ be a collection of all possible partitions of X, and hence $\Pi \in \mathcal{P}$. Given a partition $\Pi \in \mathcal{P}$ and $x \in X$, let $C_\Pi(x)$ be the cluster of Π containing x.

Definition 2.3 (Gupta et al. [2003]). *An* (r, ε)-padded probabilistic ball-partition *of a metric* (X, d) *is a probability distribution* μ *over* $\mathcal{P}$ *satisfying:*

(*1*) **(bounded radius)** *Each* Π *in the support of* μ *is an* r*-ball partition.*

(*2*) **(padding)** $\forall x \in X$, $\Pr_\mu \left[d(x, X \setminus C_\Pi(x)) \geq \varepsilon r \right] \geq \frac{1}{2}$.

(This is called a padded probabilistic decomposition in Gupta et al. [2003].) Each cluster C in every partition Π in the support of a probabilistic ball-partition μ has radius at most r; and for any $x \in X$, a random r-ball partition Π drawn from the distribution μ does not cut $\mathbf{B}(x, \varepsilon r)$ (and hence $\mathbf{B}(x, \varepsilon r)$ is contained in cluster $C_\Pi(x) \in \Pi$) with probability $\geq 1/2$.

2.3 Padded Probabilistic Hierarchical Decompositions

In this section, we define a (ρ, ε)-padded probabilistic hierarchical decomposition (PPHD) of the metric (X, d), on which the routing algorithm is based. A PPHD is a probability distribution over HDs that has a "probabilistic padding" property similar to that in Definition 2.3. For any pair of nodes s, t in X and any ball containing both s and t with a diameter of $\approx d(s,t)$, the PPHD ensures that this ball is contained in a single cluster of radius only slightly ($\approx \alpha$ factor) larger than $d(s,t)$ at a suitable level with probability $\geq \frac{1}{2}$. Thus the shortest s-t path is contained entirely in this cluster of radius not much more than $d(s,t)$. This is the general intuition for PPHDs and the starting point for the routing algorithm.

For our applications, we refine PPHDs so that they consist of only $m = O(\alpha \log \alpha)$ of HDs. We first give an existence proof, using the Lovász Local Lemma (LLL), to show that such decompositions exist in Section 2.3.1. We then outline a randomized polynomial-time algorithm to find the decompositions using Beck's techniques Beck [1991] in Section 2.3.2.

The existence proof for the PPHDs has the following outline. We first give a randomized algorithm to form a single random hierarchical decomposition $\mathbf{\Pi}$, which proves the existence of PPHDs, albeit with support over an exponential number of HDs. To reduce the size to something that depends only on α, we have to use the locality property of the metric space and the LLL. One significant complication in the proof is that we cannot use the standard top-down decomposition schemes to construct PPHDs, since they have long-range correlations that preclude the application of the LLL. Our solution to this problem is to build the decomposition trees in a bottom-up fashion and to make sure that the coarser partitions respect the cluster boundaries made in the finer partitions.

2.3.1 Existence of PPHDs

Motivated by the routing application, we are interested in finding the following structure, which we call a (ρ, ε)-*padded probabilistic hierarchical decomposition*. This is a probability distribution μ over ρ-hierarchical decompositions (as defined in Definition 2.2) so that given $\mathbf{B}(x, \varepsilon r)$ with $r \approx \rho^i$, if we choose a random ρ-HD $\mathbf{\Pi}$ from μ and examine the partition Π_i in it, $\mathbf{B}(x,r)$ is cut in this partition Π_i with probability at most $\frac{1}{2}$.

Definition 2.4 (PPHD). *A* (ρ, ε)-padded probabilistic hierarchical decomposition *(referred to as a* (ρ, ε)*-PPHD) is a distribution* μ *over* ρ*-hierarchical decomposi-*

tions, such that for any point $x \in X$ and any value r s.t. $\rho^{i-1} \le r \le \rho^i$,

$$\Pr_{\mathbf{\Pi} \in \mu}[\mathbf{B}(x, \varepsilon r) \text{ is cut in } \Pi_i] \le \tfrac{1}{2},$$

where the random ρ*-hierarchical decomposition chosen is* $\mathbf{\Pi} = (\Pi_i)_{i=0}^h$. *The* degree *of the PPHD* μ *is defined to be* $\deg(\mu) = \max_{\mathbf{\Pi} \in \mu} \deg(\mathbf{\Pi})$.

Note that the definition of a PPHD extends both the idea of a padded probabilistic ball-partition and that of HDs—we ask for a distribution over entire HDs, instead of over ball-partitions at a certain scale r. However, having picked a random ρ-HD $\mathbf{\Pi} = (\Pi_i)_{i=0}^h$ from this distribution, we demand that balls of radius $\approx \varepsilon\rho^i$ be cut with small probability only in partition Π_i that is "at the correct distance scale". Our main theorem of this section is the following:

Theorem 2.2. *Given a metric* (X,d)*, there exists a* (ρ,ε)*-PPHD* μ *for* (X,d) *with* $\rho = O(\alpha)$ *and* $\varepsilon = O(1/\alpha)$*. The degree* $\deg(\mu)$ *of the PPHD is at most* $\alpha^{O(\alpha)}$*. Furthermore, there exists a distribution* μ_m *whose support is over only* $m = O(\alpha \log \alpha)$ *HDs.*

Since any hierarchical decomposition $\mathbf{\Pi}$ can be associated with a tree $T_{\mathbf{\Pi}}$ (as mentioned in Section 2.2.1), the above theorem can be viewed as guaranteeing a set of m trees such that the level-i clusters in half of these trees do not cut a given ball of radius $\approx \varepsilon\rho^i$.

We prove Theorem 2.2 in the rest of this section. We first prove in Theorem 2.3 that one can obtain the result where the PPHD μ has support over many HDs. We then use the Lovász Local Lemma to show that a PPHD distribution μ_m with support over only a small number of HDs exists.

Padded Probabilistic Hierarchical Partitions. If we do not care about the number of HDs in the support of a PPHD, the existence result of Theorem 2.2 has been proved earlier Talwar [2004] with better guarantees; the proof basically follows from the padded decompositions given in Gupta et al. [2003]. However, we now give another proof that introduces ideas that are ultimately useful in obtaining a PPHD distribution whose support is over a small number of HDs.

Theorem 2.3. *Given a metric* (X,d)*, there exists a* (ρ,ε)*-PPHD* μ *for* (X,d) *with* $\rho = O(\alpha)$ *and* $\varepsilon = O(1/\alpha)$*, and with degree* $\deg(\mu) = \alpha^{O(\alpha)}$*. Furthermore, one can sample from* μ *in polynomial time.*

Proof. We define a randomized process that builds a random hierarchical decomposition tree in a bottom-up fashion, instead of the usual top-down way. To build a HD $\mathbf{\Pi}$, we start with $(\Pi_0 = \{\{x\} : x \in X\})$ and perform an inductive step. At any step, we are given a partial structure $(\Pi_i, \ldots, \Pi_0)$ where for each $j \leq i$, the clusters in Π_{j-1} (which is an η_{j-1}-ball partition) are contained within the clusters of Π_j. We then build a new partition Π_{i+1}, with all clusters of Π_i being contained within clusters of Π_{i+1}. We have to ensure that clusters of Π_{i+1} are contained in balls of radius at most η_{i+1} and that any ball of radius εr for $\rho^i \leq r \leq \rho^{i+1}$ is cut in Π_{i+1} with probability at most $\frac{1}{2}$. This way, we end up with a valid random HD $\mathbf{\Pi}$. The claimed probability distribution μ is the one naturally generated by this algorithm. To create the clusters of Π_{i+1}, we use a decomposition procedure whose property is summarized in the following lemma.

---

0. Let $Y \leftarrow X$, $p \leftarrow \frac{c\alpha\Gamma}{\Lambda}$ for constant c to be fixed later, N be a $\Lambda/2$-net of X.
1. Pick an arbitrary "root" vertex $v \in N$ not picked before
2. Set the initial value of the "radius" $L \leftarrow \Lambda/2$
3. Flip a coin with bias p
4. If the coin comes up heads, goto Step 11
5. If the coin comes up tails, increment L by Γ
6. If $L > \Lambda(1 - 1/4\alpha)$
7.   choose a value $\hat{L}$ from $[0, \Lambda/(4\alpha)]$ u.a.r.
8.   round down $\hat{L}$ to the nearest multiple of Γ
9.   set $L \leftarrow \Lambda(1 - 1/4\alpha) + \hat{L}$
10. Else goto Step 3
11. Form a new cluster C' in Π'' containing all clusters in $\Pi' \cap Y$ with centers lie in $\mathbf{B}(v, L)$
12. Remove the vertices in C' from Y
13. (Remark: C' has radius at most $\Lambda + \Gamma$)
14. If $Y \neq \emptyset$ goto Step 1
15. End

Figure 2.3.1. **Algorithm** CUT-CLUSTERS

Lemma 2.1. *Given a metric (X,d) with a Γ-ball partition Π' of X into clusters lying in balls of radius at most $\Gamma \geq 1$, and a value $\Lambda \geq 8\Gamma$, there is a randomized algorithm to create a $(\Lambda+\Gamma)$-ball partition Π'' of X, where each cluster of Π' is contained in some cluster of Π'', and for any $x \in X$ and radius $0 \leq r \leq \Lambda$,*

$$\Pr[\mathbf{B}(x,r) \textit{ is cut in } \Pi''] \leq \frac{O(r+\Gamma)}{\Lambda}\alpha.$$

Proof. Note that we can assume that $\Gamma < \Lambda/c\alpha$ and $\Lambda \geq \alpha$, since otherwise the lemma is trivially true. Using the algorithmCUT-CLUSTERS given in Figure 2.3.1, we create a partition of Y (and hence of X); all distances are measured according to the original distance function d in X.

Let us define $\mathcal{B}_x = \mathbf{B}(x,r)$. Note that if $\mathcal{B}_x$ is cut in Π'' due to some value of L from $v \in N$ (for the first time), then L falls into the interval $[d(v,x)-r-\Gamma, d(v,x)+r+\Gamma]$. Indeed, if $\mathcal{B}_x$ is cut in Π'', there are at least two clusters $C_1', C_2' \in \Pi'$ such that they both cut $\mathcal{B}_x$, and $\mathbf{B}(v,L)$ contains one of their centers but not both. Since both clusters intersect $\mathcal{B}_x$, their centers c_1' and c_2' are at distance at most $r+\Gamma$ from x. If $L < d(v,x)-r-\Gamma$, the triangle inequality implies that $\mathbf{B}(v,L)$ cannot contain either center. Similarly, if $L > d(v,x)+r+\Gamma$, $\mathbf{B}(v,L)$ contains both of them. Hence the value of L must fall into the interval indicated above.

If a cut in Step 11-12 is made due to the appearance of a heads in Step 4, we call such a cut a *normal cut*; else we call it a *forced cut*. We now bound the probability that the ball $\mathcal{B}_x = \mathbf{B}(x,r)$ is cut due to either type.

Normal cuts. Consider the first instant in time when the parameter L for some root $v \in N$ reaches a value such that the cut obtained by taking all $\Pi' \cap Y$ clusters with centers in $B(v,L)$ would cut $\mathcal{B}_x$. (If there is no such time, then $\mathcal{B}_x$ is never cut by a normal cut.) In this case, L must also be in the range $d(v,x) \pm (r+\Gamma)$, and increases with time. Now either (i) we make a normal cut before L goes outside this range; or (ii) we make a forced cut; or (iii) L goes outside the range and we make no cut in this range. In any case, the fate of $\mathcal{B}_x$ is decided; $\mathcal{B}_x$ is either cut or contained in a new cluster with center v. We now upper-bound the probability that event (i) happens. There are at most $2(r+\Gamma)/\Gamma$ coin flips made (with bias p) when the value of L is in the correct range of width at most $2(r+\Gamma)$ and one of these flips must come up heads for the cut to be made. The trivial union bound now shows this probability to be at most $\frac{2(r+\Gamma)}{\Gamma}\,p = \frac{2c(r+\Gamma)}{\Lambda}\alpha$.

Forced cuts. Let us look at some root $v \in N$ and bound the probability that a forced cut is made with cutting radius L from v in some range $\mathcal{R}_x = d(v,x) \pm (r+\Gamma)$. Since the cut is forced and the value of L is greater than $\Lambda(1-1/4\alpha) \geq 3\Lambda/4$, we must have flipped a sequence of at least $\Lambda/4\Gamma$ successive tails; the probability of this event is at most

$$(1-p)^{(\Lambda/4\Gamma)} \leq e^{-p\Lambda/4\Gamma} = e^{-\frac{c}{4}\alpha}. \tag{2.3.1}$$

Now, we choose $\hat{L}$ to be a multiple of Γ uniformly in a range of width at most $\Lambda/4\alpha$, and hence the probability that L falls into a range of length $2(r+\Gamma)$ is at most $2(r+\Gamma)/(\Lambda/4\alpha)$. Multiplying this by (2.3.1), we obtain a bound of $e^{-\frac{c}{4}\alpha} \times \frac{8(r+\Gamma)}{\Lambda}\alpha$ on the probability that a forced cut is made around v with L in the range $\mathcal{R}_x$ such that the cluster C' with center v in Π'' may cut $\mathcal{B}_x$. Finally, for any $x \in X$, $\mathcal{B}_x$ can only be cut by clusters from roots $v \in N$ that are at distance at most $(r+\Gamma)+\Lambda \leq 3\Lambda$ from x; by Prop. 2.1, there are at most $|\mathbf{B}(x,3\Lambda) \cap N| = (\frac{6\Lambda}{\Lambda/2})^{\alpha} \leq (12)^{\alpha}$ of such roots. Now we choose c to be large enough; the probability of B_x being cut by a forced due to any such root is at most $12^{\alpha} \times e^{-\frac{c}{4}\alpha} \times \frac{8(r+\Gamma)}{\Lambda}\alpha \leq \frac{O(r+\Gamma)}{\Lambda}\alpha$ by the union bound. □

We now use the above lemma to prove Theorem 2.3. Using $\Pi' = \Pi_i$, $\Gamma = \eta_i < \rho^i(\rho/(\rho-1))$, and $\Lambda = \eta_{i+1} - \Gamma = \rho^{i+1}$, and using $N = N_{i+1}$ (which is a $\rho^{i+1}/2 = \Lambda/2$ net), we create a $(\Gamma+\Lambda = \eta_{i+1})$-ball partition such that for all x and all $r \leq \rho^{i+1}$ and $\varepsilon = O(1/\alpha)$, we have

$$\Pr[\mathbf{B}(x,\varepsilon r) \text{ cut}] \leq \frac{O(\varepsilon r+\Gamma)}{\Lambda}\alpha \leq \frac{O(\rho^i)}{\rho^{i+1}}\alpha \leq \frac{1}{10} < \frac{1}{2}, \tag{2.3.2}$$

for ρ/α and c being large enough constants. The probability distribution μ over all decompositions $\mathbf{\Pi}$ thus generated satisfy the requirements of a PPHD as given in Definition 2.4. Finally, we bound the degree $\deg(\mu)$ of the PPHD μ; note that each level-i cluster is centered at some $v \in N_i$, hence the number of level-i clusters contained in some level-$(i+1)$ cluster is $(2\eta_{i+1}/(\rho^i/2))^{O(\alpha)} = \alpha^{O(\alpha)}$ by Prop. 2.1. □

Few Hierarchical Decompositions. The above proof immediately gives us a

PPHD μ_M with a support on only $M = O(\log n + \log\log\Delta)$ HDs. By sampling from the distribution μ for M times, we get the HDs $\mathbf{\Pi}^{(1)}, \dots, \mathbf{\Pi}^{(M)}$, and let the PPHD μ_M be the uniform distribution on these HDs. By (2.3.2), for each $j \in [1 \dots M]$, point $x \in X$ and radius $r \leq \rho^i$, $\mathbf{B}(x, \varepsilon r)$ is not cut in the partition $\Pi_i^{(j)}$ with probability $1/10$; hence a Chernoff bound implies that this ball is cut in the level-i partitions of more than $M/2$ of the HDs with probability less than $1/(n\log\Delta)^{O(1)}$. Now taking the trivial union bound over all possible values of the center $x \in X$, and all the $\log\Delta$ values of r which are powers of 2 shows that the μ_M is a $(\rho, \varepsilon/2)$-PPHD whp.

Even Fewer Hierarchical Decompositions. While the proof of Theorem 2.3 and the discussion above do not produce a PPHD with small support (of size $O(\alpha \log \alpha)$), we have seen all the essential ideas required to prove the existence of such a distribution μ_m and hence to complete the proof of Theorem 2.2. To prove this result, we use the locality of the construction, in conjunction with the Lovász Local Lemma (LLL). This locality property is the very reason why we built the hierarchical decomposition bottom-up; it ensures that if any particular ball is not cut at some low level i (the "local decisions"), it is not cut at levels higher than i (i.e., the "non-local decisions"). Also, we choose the decomposition procedure of Theorem 2.1 in preference to others (e.g., those in Gupta et al. [2003] and Talwar [2004]) since they choose a single random radius for all clusters in one particular partition Π of X, which causes correlations across the entire metric space.

Proof of Theorem 2.2: To show that there is a distribution μ_m over only $m = O(\alpha\log\alpha)$ trees, we use an idea similar to that in the previous section, augmented with some ideas from Gupta et al. [2003]. Instead of building one hierarchical decomposition $\mathbf{\Pi}$ bottom-up, we build m hierarchical decompositions $\mathbf{\Pi}^{(1)}, \dots, \mathbf{\Pi}^{(m)}$ simultaneously (also from the bottom up).

As before, the proof proceeds inductively; we assume that we are given level-i partitions $\Pi_i^{(1)}, \dots, \Pi_i^{(m)}$, where $\Pi_i^{(j)}$ is the level-i partition belonging to $\mathbf{\Pi}^{(j)}$. We then show that we can build level-$(i+1)$ partitions $\Pi_{i+1}^{(1)}, \dots, \Pi_{i+1}^{(m)}$ where each $\Pi_i^{(j)}$ is a refinement of the corresponding $\Pi_{i+1}^{(j)}$, and any given ball $\mathbf{B}(x, \varepsilon r)$ with $\rho^i \leq r \leq \rho^{i+1}$ is cut in at most $m/2$ of these level-$(i+1)$ partitions. We start off this process with each $\Pi_0^{(j)} = \{\{x\} : x \in X\}$ being the partition consisting of all singleton points in X. Let $J = \{1, \dots, m\}$. Given m level-i partitions $(\Pi_i^{(j)})_{j \in J}$, we create m level-$(i+$

1) partitions $(\Pi_{i+1}^{(j)})_{j\in J}$ using the procedure in Lemma 2.1 independently on each of the m decompositions; parameters are set as in the proof of Theorem 2.3, with $\Lambda = \rho^{i+1}$, $\Gamma = \eta_i$, and $\varepsilon = 1/O(\alpha)$. This extends the m hierarchical decompositions to the $(i+1)^{st}$ level; it remains to show that the probability of balls being cut is small.

To describe the events of interest, let us take $\beta = \varepsilon\rho^{i+1}$ and define Z to be a β-net of X. For each $z \in Z$, define $\mathcal{B}_z$ to be $\mathbf{B}(z, 2\beta)$, and $\mathcal{E}_z^{i+1}$ to be event that $\mathcal{B}_z$ is cut in more than $m/2$ of the partitions $(\Pi_{i+1}^{(j)})_{j=1}^m$, which we refer to as a "bad" event (used in Section 2.3.2). We prove the claim using the Lovász Local Lemma.

Claim 2.1. *Given any* $(\Pi_i^{(j)})_{j=1}^m$, $\Pr[\bigwedge_{z\in Z} \overline{\mathcal{E}_z^{i+1}}] > 0$.

Lemma 2.2 (Lovász Local Lemma). *Given a set of events* $\{\mathcal{E}_z^{i+1}\}_{z\in Z}$*, suppose that each event is mutually independent of all but at most* B *other events. Further suppose that, for each event* $\mathcal{E}_z^{i+1}$*,* $\Pr[\mathcal{E}_z^{i+1}] \le p$*. Then if* $\mathrm{e}p(B+1) < 1$*,* $\Pr[\bigwedge_{z\in Z} \overline{\mathcal{E}_z^{i+1}}] > 0$.

Proof of Claim 2.1: First, let us calculate the probability of $\mathcal{E}_z^{i+1}$: by changing the constant in ε, we can make the probability that a ball $\mathcal{B}_z$ is cut in one level-$(i+1)$ partition to be at most $1/8$. Let us denote by A_z^j the event that $\mathcal{B}_z$ is cut in partition $\Pi_{i+1}^{(j)}$. The expected number of partitions in which the ball is cut is at most $m/8$. Since the partitions are constructed independently, the probability for the event $\mathcal{E}_z^{i+1}$ that $\mathcal{B}_z$ is cut in $m/2$ partitions (which is at least four times the expectation) is at most $\exp(-9m/40)$; this can be established using a standard Chernoff bound. This, in turn, is at most $(0.8)^m$, which we define to be p.

Next we show that an event $\mathcal{E}_z^{i+1}$ is mutually independent of all events $\mathcal{E}_{z'}^{i+1}$ such that $d(z,z') > 4\eta_{i+1}$. For each partition $\Pi_{i+1}^{(j)}$, each root $v \in N_{i+1}$ determines its radius by conducting a random experiment independent of any other roots' experiments. These random experiments, and only these, determine whether events such as A_z^j occur. In turn, whether event $\mathcal{E}_z^{i+1}$ occurs is determined only by events $A_z^1, \ldots, A_z^m$. For a particular j, for each z, all of the cuts that could affect $\mathcal{B}_z$ in the algorithm CUT-CLUSTERS are made from roots $v \in N_{i+1}$ at distance at most $2\beta + \Gamma + \Lambda = 2\beta + \eta_{i+1} < 2\eta_{i+1}$ from z. Whether event A_z^j occurs is determined by the experiments corresponding to these roots alone. If $d(z,z') > 4\eta_{i+1}$, then there is no intersection between the experiments for z and the experiments for z'. Since $\mathcal{E}_z^{i+1}$ is determined by $A_z^1, \ldots, A_z^m$, $\mathcal{E}_z^{i+1}$ is mutually independent of the set of all $\mathcal{E}_{z'}^{i+1}$ such that $d(z,z') > 4\eta_{i+1}$.

We apply the LLL now. Note that the number of $z' \in Z$ within distance $4\eta_{i+1}$ of $\mathcal{E}_z^{i+1}$ for $z \in Z$ is at most $|\mathbf{B}(z, 4\eta_{i+1}) \cap Z| \leq \left(\frac{8\eta_{i+1}}{\beta}\right)^{\alpha} \leq O(\alpha)^{\alpha}$. We define this quantity to be B; $ep(B+1)$ is at most 1 for $m = O(\alpha \log \alpha)$ and Claim 2.1 follows. $\square$

Having proved the claim, let us now show that with nonzero probability, each $\mathbf{B}(x,r)$ for $x \in X$ and $\rho^i \leq r \leq \rho^{i+1}$ is not cut in at least $m/2$ of the level-$(i+1)$ partitions $(\Pi_{i+1}^{(j)})_{j \in J}$. Let us call this event SC_{i+1}. The claim shows that with nonzero probability, each ball $\mathcal{B}_z$ with $z \in Z$ is not cut in at least $m/2$ of the partitions $(\Pi_{i+1}^{(j)})_{j \in J}$. Since each $x \in X$ is at distance at most β to some $z_x \in Z$, the triangle inequality implies that $\mathbf{B}(x, \varepsilon r) \subseteq \mathbf{B}(x, \beta)$ is not cut if $\mathbf{B}(z_x, 2\beta)$ is not cut, which holds in at least half of the partitions. Hence SC_{i+1} also holds with nonzero probability.

Finally, we prove that we can choose a random set of HD's $(\mathbf{\Pi}^{(j)})_{j \in J}$ such that SC_{i+1} occurs for each $1 \leq i+1 \leq h$ *simultaneously* with nonzero probability. The key to the proof is that we have assumed an arbitrary (worst-case) set of partitions $(\Pi_i^{(j)})_{j=1}^m$ at level i in proving a nonzero lower bound on $\Pr[SC_{i+1}]$. Hence, we can ignore any dependence among the events SC_{i+1} for $1 \leq i+1 \leq h$, and simply multiply their nonzero probabilities together to obtain a nonzero lower bound on the probability that they all occur simultaneously. $\square$

2.3.2 An Algorithm for Finding the Decompositions

The above procedure can be made algorithmic using an approach based on Beck's algorithmic version of the LLL (see, e.g., Alon and Spencer [1992]; Beck [1991]). The decomposition satisfies all properties of the one that is shown to exist using LLL in Theorem 2.2, although with some changes in constant parameter values. As in the proof of Theorem 2.2, we build $m = O(\alpha \log \alpha)$ HDs level by level in a bottom-up fashion.

On any particular level $i+1$, we begin by choosing m partitions at random. After making the random choices, we examine the partitions and identify all of the bad events that have occurred. We then group together bad events that may depend on each other, as well as "good" events that may depend on the bad events. Each group forms a connected component in the LLL dependency graph. We show that, with high probability, all connected components have size $O(\log \nu)$, where $\nu = |Z|$ is the size of the $\varepsilon\rho^{i+1}$-net of X.

Once the groups have been identified, we need to eliminate the bad events. Hence, for each group, we "undo" all of the random choices concerning that group, while not modifying any choices that do not affect the group. New choices must be made for each group so that no bad event occurs. Because the group size is small (the number of centers $v \in N_{i+1}$ concerning the group that we choose random radius for is also $O(\log \nu)$), we can find new settings for these choices using exhaustive search in polynomial time.

One interesting complication in this proof is that the set of clusters containing a group have different shapes in the m different partitions. In each partition, we cut out a "hole", and redo the choices within the hole. The boundary of the hole is formed from the boundaries of the clusters that may influence the bad events (and the good events) in the group. In forming the boundary, additional good events may be added to the hole. As a consequence, it is possible that a good event inside a hole in one partition may appear inside a different hole in another partition. Hence, when we perform exhaustive search, these holes must be considered together. However, our method of bounding the size of each connected component already takes into account any merging of holes on account of shared good events, so that we never have to redo the choices for a group of size more than $O(\log \nu)$.

Another issue is that the subset of centers in a hole that belong to N_{i+1}, the $\rho^{i+1}/2$-net that covers the entire metric, may not by themselves cover the hole. (Portions of the hole may be covered by centers outside the hole.) So for each of the m partitions, we may have to add additional net points inside the hole to obtain a complete cover for it. We show that the size of net points in the hole increases by only a constant factor and remains $O(\log \nu)$, and the degree of the hierarchical decomposition trees is at most $\alpha^{O(\alpha)}$ as before.

2.4 The $(1+\tau)$-Stretch Routing Schemes

Given a (ρ,ε)-PPHD μ_m with a support on m HDs, we can now define, for every $0<\tau\leq 1$, a $(1+\tau)$-stretch routing scheme which uses routing tables of size at most $m(\alpha/\tau)^{O(\alpha)}\log\Delta\log\delta$ bits at every node.

We consider routing schemes in two models. In a basic model, we assume that there is no underlying routing fabric and each node can only send packets to its direct neighbors. In a second model, we can build an overlay hierarchical routing scheme upon an underlying routing fabric like IP that can send packets to any specific node in the network. We specify the routing algorithm in the basic model, but also indicate how one can circumvent certain steps of this algorithm when an underlying routing mechanism is given.

Let us recall some of the notation defined earlier. Let $(\mathbf{\Pi}^{(j)})_{j=1}^{m}$ be the m hierarchical decompositions on which μ_m has positive support, and the level-i partition corresponding to $\mathbf{\Pi}^{(j)}$ be called $\Pi_i^{(j)}$. Recall that we can associate each hierarchical decomposition $\mathbf{\Pi}^{(j)}$ with a tree T_j (as outlined in Section 2.2.1). Note that each of these trees has a $\deg(\mu_m)$ bounded by $\alpha^{O(\alpha)}$ and a height of at most $h=\lceil\log_\rho\Delta\rceil$. Recall that each internal vertex of the tree T_j at level i corresponds to a cluster of $\Pi_i^{(j)}$ and leaves of $T_j, \forall j\in J$, correspond to vertices in X, where $J=\{1,\ldots,m\}$. Let each internal vertex v of each tree T_j *label* its children by numbers between 1 and $\deg(\mu_m)$; v does not label anything with the number 0, but uses it to refer to its parent. Note that this allows us to represent any path in a tree T_j by a sequence of at most $2h=O(\log_\rho\Delta)$ labels.

2.4.1 The Addressing Scheme

Given a tree T_j and a vertex $x\in X$, we assign x a *local address* $\mathsf{addr}_j(x)$, which consists of $h=\lceil\log_\rho\Delta\rceil$ blocks, one for each level of the tree T_j. Each block has a fixed length. The i^{th} *block* of the $\mathsf{addr}_j(x)$ corresponds to partition $\Pi_i^{(j)}$ and contains the label assigned to the cluster C_x containing x in $\Pi_i^{(j)}$ by C_x's parent in T_j. Since any such label is just a number between 1 and $\deg(\mu_m)$, where $\deg(\mu_m)=\alpha^{O(\alpha)}$, we need $O(\alpha\log\alpha)$ bits per block. In fact, one can extend this addressing scheme to any cluster C in T_j. If C is a level-i cluster, the k^{th}-block of $\mathsf{addr}_j(C)$ contains $*$'s for $k < i$; $\mathsf{addr}_j(X)$ for the root cluster of T_j contains all $*$'s matching all vertices in X.

The *global address* $\mathsf{addr}(x)$ of point $x \in X$ is the concatenation $\langle \mathsf{addr}_1(x), \cdots, \mathsf{addr}_m(x) \rangle$ of its local addresses $\mathsf{addr}_j(x)$ for $j \in J$. Since each cluster C belongs to only one tree T_j, we define $\mathsf{addr}_{j'}(C)$ to be a sequence of #'s of the correct length (where # are dummy symbols matching nothing), and hence define a global address of C as well. (This is only for simplicity; in actual implementations, cluster addresses for T_j can be given by the tuple $\langle \mathsf{addr}_j(C), j \rangle$.)

Since there are $O(\alpha \log \alpha)$ bits per block, h blocks per local address, and m local addresses per global address, substitution of the appropriate values gives the address length A to be at most $m \times h \times \lceil \log(\deg(\mu_m)) \rceil = O(\alpha \log \alpha) \times \left\lceil \log_\rho \Delta \right\rceil \times O(\alpha \log \alpha) = O(\alpha^2 \log \alpha \log \Delta)$ bits.

2.4.2 The Routing Table

For each point $x \in X$, we maintain a routing table Route_x that contains the following information for each T_j, $1 \le j \le m$:

(1) For each ancestor of x in T_j that corresponds to a cluster C containing x, we maintain a table entry for C.

(2) Moreover, for each such C, we maintain an entry for each descendant of C in T_j reachable within ℓ hops in tree T_j. Here $\ell = \Theta(\log_\rho 1/\varepsilon\tau)$, with the constants chosen such that $\eta_{i-\ell} \le \frac{\varepsilon\tau}{4} \rho^{i-1}$.

In the routing table Route_x for x, each of the above entries thus corresponds to some level-i' cluster C' in T_j. Let $\mathsf{close}_x(C')$ be the closest point in C' to x. (We assume, w.l.o.g., that ties are broken in some consistent way, so that any node y on a shortest path from x to $\mathsf{close}_x(C')$ has the value $\mathsf{close}_y(C') = \mathsf{close}_x(C')$; in fact, this consistency is the only property we use.) For this C', Route_x stores **(a)** the global address $\mathsf{addr}(C')$ by which the table is indexed, **(b)** the identity of a "next hop" neighbor y of x that stays on a shortest path from x to the closest point $\mathsf{close}_x(C')$ in C', and **(c)** an extra bit $\mathsf{ValidPath}_x(C')$: if the cluster ℓ levels above C' in T_j is the cluster C, then $\mathsf{ValidPath}_x(C')$ is set to be `true` if $\mathbf{B}(x, \varepsilon\rho^{i'+\ell})$ is entirely contained within cluster C and $d(x, \mathsf{close}_x(C')) \le \varepsilon\rho^{i'+\ell}$, and is set to be `false` otherwise. Of course, if we reach the root of T_j while trying to go up ℓ

levels, then the bit is set to be `true`. Note that if there is an underlying routing fabric like IP, we can store the IP-address of some node in C' (say, the closest one) instead of **(b)** and **(c)** above.

Lemma 2.3. *The number of entries in the routing table* Route_x *of any* $x \in X$ *is at most* $\log\Delta \times (\alpha/\tau)^{O(\alpha)}$.

Proof. Let us estimate the number of entries in Route_x for any $x \in X$. There are m trees. For each tree T_j, for all $j \in J$, there are $h = \lceil \log_\rho \Delta \rceil$ ancestors of x and the degree of the tree is bounded by $\deg(\mu_m) = \alpha^{O(\alpha)}$. Recall that ρ and $1/\varepsilon$ are both $O(\alpha)$, and hence $\ell = O(\log(\alpha/\tau))$. Plugging these values in, we get that the number of entries for x across m trees is at most $m \times h \times (\deg(\mu_m))^{\ell} = O(\alpha \log \alpha) \times O(\log_\alpha \Delta) \times \alpha^{O(\alpha\ell)} = \log\Delta \times (\alpha/\tau)^{O(\alpha)}$. Each entry is indexed by one global address (of at most $A = O(\alpha^2 \log\alpha \log\Delta)$ bits, which we do not store in Route_x since we can deduce it from $\mathsf{addr}(x)$ based on the clustering structure); each entry indeed contains the identity of the next hop (which uses $O(\log\delta)$ bits, where δ is the maximum degree of G), a path length field (to be specified in Section 3.2), and one additional $\mathsf{ValidPath}$ bit. □

The forwarding algorithm makes use of two functions, $\mathsf{NextHop}_x$ and $\mathsf{PrefMatch}_x$. For a point x and a level-i' cluster C' in T_j, the function $\mathsf{NextHop}_x(\mathsf{addr}(C'))$ returns the next hop on the path from x to $\mathsf{close}_x(C')$ provided that the next hop does not leave the cluster C at level $i' + \ell$ that contains C', and null otherwise. (As we shall see, the packet forwarding algorithm is guaranteed never to encounter a null next hop.) Given points x and t in X, the function $\mathsf{PrefMatch}_x(t)$ returns an $\mathsf{addr}(C')$ in Route_x such that in some T_j, t belongs to the level-i cluster C', $\mathsf{ValidPath}_x(C')$ is `true`, and the value i is *the smallest* across all trees. Note that both of these functions can be computed efficiently by node x. Furthermore, it is possible to support the functions with data structures of size comparable to that of Route_x.

Note that once the points in X have been assigned addresses (for which we have described only an off-line algorithm), the routing tables can be built up in a completely distributed fashion. In particular, a distributed breadth-first-search algorithm can be applied to determine whether a ball of a certain radius is cut in a particular decomposition, and a distributed implementation of the Bellman-Ford algorithm can be used to establish the next-hop entries for destinations for which the shortest paths lie within a certain cluster.

2.4.3 The Forwarding Algorithm

The idea behind the forwarding algorithm is to start a packet off from its origin s towards an *intermediate* cluster C containing its destination t; the packet header thus consists of two pieces of information $\langle \mathsf{addr}(t), \mathsf{addr}(C)\rangle$, where t is the destination node for the packet and C is the *intermediate* cluster containing t. Initially, the cluster can be chosen (degenerately) to be the root cluster of (say) tree T_1.

Upon reaching a node x in the intermediate cluster C, a new and smaller intermediate cluster C', also containing t, must be chosen, possibly from a different tree; the packet header must be updated with $\mathsf{addr}(C')$ that remains the same until reaching C'. Suppose that the new cluster C' containing t is at level i'. After selecting this cluster, the packet is sent off towards C' with the new header, following a shortest path that stays within the cluster $\hat{C}$ at level $i'+\ell$ that contains both x and C'. This process is repeated until ultimately the packet reaches the cluster containing only the destination t. The algorithm is presented in Figure 2.4.2.

Let packet header be $\langle \mathsf{addr}(t), \mathsf{addr}(C)\rangle$.
If C contains x, the current node, then
    find $\mathsf{addr}(C') \leftarrow \mathsf{PrefMatch}_x(t)$
    let $y \leftarrow \mathsf{NextHop}_x(\mathsf{addr}(C'))$
    forward packet with new header $\langle \mathsf{addr}(t), \mathsf{addr}(C')\rangle$ to y.
Else (now $x \notin C$)
    let $y \leftarrow \mathsf{NextHop}_x(\mathsf{addr}(C))$
    forward packet with unchanged header $\langle \mathsf{addr}(t), \mathsf{addr}(C)\rangle$ to y.
End

Figure 2.4.2. The Forwarding Algorithm at Node x

Theorem 2.4. *The forwarding algorithm has a stretch of at most* $(1+\tau)$*, where* $\tau \leq 1$.

Proof. We first show that the algorithm is indeed valid; each of the steps can be executed and the packet eventually reaches t. Suppose that the packet has just reached a node x in an intermediate cluster C containing t (with $\mathsf{addr}(C)$ in its header); thus x needs to execute Step 3 to find a new cluster C' containing t. Clearly, $\mathsf{PrefMatch}_x(t)$ can return the root cluster C_{root} of any T_j, since it contains t. We show, however, that the cluster C' returned by $\mathsf{PrefMatch}_x(t)$ has a small diameter and nodes along a valid shortest path from x to C' will forward the packet correctly until it reaches C'.

Lemma 2.4. *If the packet is at node x with distance to the target t being $d(x,t) \leq \varepsilon\rho^i$, Step 3 must return some $\mathsf{addr}(C')$ such that cluster $C' \ni t$ is at level $(i-\ell)$ or lower in some $T_{j'}$ with $\mathsf{ValidPath}_x(C')$ being $\mathtt{true}$. Furthermore, all vertex v on all shortest paths from x to $\mathsf{close}_x(C') = \mathsf{close}_v(C')$ has a non-null $\mathsf{NextHop}_v(\mathsf{addr}(C'))$.*

Proof. The (ρ,ε)-PPHD ensures that there exists at least one tree T_j such that $\mathbf{B}(x,\varepsilon\rho^i)$ is not cut in the level-i partition $\Pi_i^{(j)}$; let $\hat{C}_{cont} \in \Pi_i^{(j)}$ be the level-i cluster in T_j that contains $\mathbf{B}(x,\varepsilon\rho^i)$. Let $C_t \in \Pi_{i-\ell}^{(j)}$ be the level-$(i-\ell)$ cluster in T_j containing t. The $\mathsf{ValidPath}_x(C_t)$ bit must be $\mathtt{true}$ since $\mathbf{B}(x,\varepsilon\rho^i) \subseteq \hat{C}_{cont}$ in $\Pi_i^{(j)}$ and $d(x,\mathsf{close}_x(C_t)) \leq d(x,t) \leq \varepsilon\rho^i$; thus $\mathsf{PrefMatch}_x$ can (and may indeed) just return $\mathsf{addr}(C_t)$ given no "better" choices. However, $\mathsf{PrefMatch}_x$ always finds a cluster C' in some $T_{j'}$, at the *lowest* level across all trees, such that $t \in C'$, and $\mathsf{ValidPath}_x(C')$ is $\mathtt{true}$ in Route_x. Let the level of C' be i'; the value i' is at most $(i-\ell)$. Now Let $\hat{C} \in \Pi_{i'+\ell}^{(j')}$ be the cluster ℓ levels above $C' \in \Pi_{i'}^{(j')}$ in $T_{j'}$ that contains both x and C'. (Such $\hat{C}$ must exist at level $i'+\ell$ for $\mathsf{addr}(C')$ to be in Route_x.) We know that $\mathbf{B}(x,\varepsilon\rho^{i'+\ell}) \subseteq \hat{C}$ and $d(x,\mathsf{close}_x(C')) \leq \varepsilon\rho^{i'+\ell}$ since $\mathsf{ValidPath}_x(C')$ is $\mathtt{true}$ in Route_x. Thus all shortest paths from x to $\mathsf{close}_x(C')$ are entirely contained in $\hat{C}$. Hence, the $\mathsf{NextHop}_v(\mathsf{addr}(C'))$ pointer at any node v on one of these paths must be non-null since all shortest paths from v to $\mathsf{close}_v(C') = \mathsf{close}_x(C')$ are all contained in $\hat{C}$, the cluster ℓ levels above C' in T_j'. $\square$

It remains to bound the path stretch. Consider the case when a packet is sent from s to t. Let C' be a cluster at level $i-\ell$ returned by Step 3 of the forwarding algorithm. Note that if the level $i \leq \ell$, then $C' = \{t\}$ and we send the packet directly to t with $\tau = 0$. Using these short distances as the base case, we now do induction on the distance from s to t.

If C' is a non-trivial cluster containing t, then we go on a shortest path from s to some vertex $v = \mathsf{close}_s(C') \in C'$. Since $t \in C'$, $d(s,v) \leq d(s,t)$. Because the diameter of C' is at most $2\eta_{i-\ell}$, $d(v,t) \leq 2\eta_{i-\ell} < \varepsilon\rho^{i-1} < d(s,t)$. (The last inequality holds because if $\varepsilon\rho^{i-1} \geq d(s,t)$, then $\mathsf{PrefMatch}_s$ would have returned a cluster at a level lower than that of C' by Lemma 2.4.) Hence, we can apply the induction hypothesis to find a path from v to t of length at most $(1+\tau)d(v,t) \leq (1+\tau)2\eta_{i-\ell}$. The path from s to t as derived from Route_s is of length at most $d(s,v) + (1+\tau)d(v,t) < d(s,t) + (1+\tau)2\eta_{i-\ell}$. The stretch of the path from s is t is then $1 + (1+\tau)2\eta_{i-\ell}/d(s,t)$. This quantity is at most $1+\tau$ since $\tau \leq 1$ and we have chosen constants so that $\eta_{i-\ell} \leq \tau\varepsilon\rho^{i-1}/4$. $\square$

3 Routing Table Construction Using Bellman-Ford

3.1 Introduction

The hierarchical routing scheme we are going to describe in this section is a completion of what is lacking in Section 2.4; hence we focus primarily on the process of building up routing tables using a distributed implementation of Bellman-Ford algorithm for the base model that we introduce in Section 3.2. For overlay routing, we store the IP address of an intermediate node to reach each destination in the routing tables and the process of routing table updates are similar to that of prefix routing, e.g., in Hildrum et al. [2002]. Although the Forwarding algorithm remains the same as that in Section 2.4.3, we will elaborate in more details on its behavior in Section 3.3 when it is coupled with the new routing algorithm.

Our routing scheme is similar in spirit to that of Closest Entry Routing (CER) scheme described in KK(Kleinrock and Kamoun [1977]). They define a hierarchical routing scheme by first specifying an "optimal" underlying hierarchical clustering structure that they impose on the network nodes, where the optimization objective is to minimize the routing table length; each level-k cluster is defined recursively as a set of level-$(k-1)$ clusters, with the level-0 clusters being individual nodes. This leads naturally to a tree representation as shown in Figure 3.1.1, where internal tree nodes represent clusters; Table 3.1 shows that the destination addresses in the routing table of node A corresponds to clusters at different levels of the decomposition tree, hence reflecting the structure of the hierarchical clustering of network nodes. In KK, two nodes share common routing table entries for all the clusters that contain both of them. KK assumes that all clusters at the same level have the same number of sub-clusters within them, and each cluster is a connected component. The KK hierarchical routing procedure leads a message down a tree path, fixing more prefix digits at each step, much as prefix routing, traversing smaller and smaller clusters that contain the destination node until it reaches the destination itself.

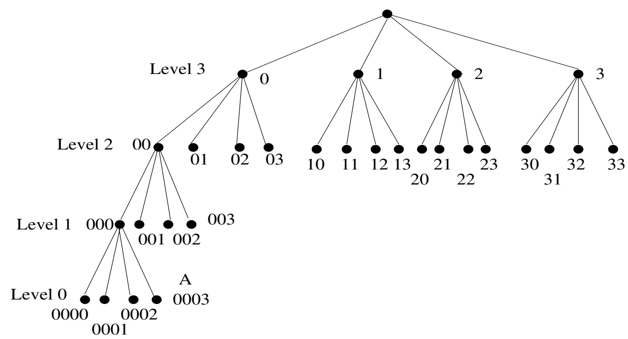

Figure 3.1.1. A 4-level Hierarchical Clustering Structure of Network Nodes

Level 3	0***	1***	2***	3***
Level 2	00**	01**	02**	03**
Level 1	000*	001*	002*	003*
Level 0	0000	0001	0002	0003

Table 3.1. Routing Table Entries in Node A in Figure 3.1.1

The reduction of routing table size generally leads to an increase in network path length. In order to derive bounds on the increase in the average path length, they further assume that a shortest-path between two nodes in a cluster lies within the cluster. They also prescribe an upper-bound of d_k on the (strong) diameter of a k^{th} level cluster, with d_k decreasing as k decreases. They show that routing schemes based on the hierarchical clustering model cause essentially no increase in the *average network path length* for a family of large distributed networks. Specifically, the networks they consider are all connected graphs upon which it is possible to fit a hierarchical clustering whose outcome satisfies the assumptions above. In addition, (a) the resulting clusters at any level satisfy the following: the diameter of any cluster S chosen is bounded above by $O(|S|^{\nu})$ for some constant $\nu \in [0,1]$, and (b) the average distance between nodes in the network is $\Theta(N^{\nu})$, where N is the size of such a network.

In contrast, our hierarchical routing schemes give bounds on the path stretch on a *per node-pair* level on certain networks that are connected graphs G, where the natural metric (X,d) induced by shortest path distances between any pair of nodes in G is a doubling metric. In addition, the main improvement our work over that of KK is: while the KK routing scheme is based on assumptions regarding the existence of a "good" partition of the network, the method itself does not provide an algorithm for computing such a partition; we are able to prove the existence of a (ρ,ε)-PPHD with a support on m Hierarchical Decompositions and actually find them by following the Clustering algorithm and its constructive algorithm described in Section 2.3. Note that while we guarantee a degree bound for the decomposition trees across all levels, we do not require they are exactly the same.

It would be ideal if once we construct such a set of network partitions, we can run the hierarchical routing algorithm specified in KK at each individual decomposition tree. However, it is not possible to directly apply KK's routing scheme or their proof techniques for three reasons. First, while KK assumes that each cluster subnetwork is fully connected, this is not satisfied in our decomposition. Second, the shortest paths between two nodes in a cluster are not guaranteed to stay within the cluster. Finally, although the maximal distance in G between vertices of C_k, for all $0 \le k \le h$, is bounded within the diameter of C_k, $2\eta_k$, which is geometrically decreasing as k decreases, it is a weak diameter bound and not necessarily satisfied by the distance induced by the subgraph corresponding to each cluster C_k.

We thus adopt as many definitions and notation as possible from KK in this section while inventing some new techniques for addressing the above issues in the design and specification of a modified hierarchical routing scheme given a (ρ,ε)-PPHD μ_m with a support on m HDs and in the analysis of the characteristics of paths as induced by the routing tables thus created. The important property of a (ρ,ε)-PPHD that we will use in defining our routing scheme is that, for $\rho^{i-1} < r \le \rho^{i}$, there is at least one tree T_j such that $\mathbf{B}(s,\varepsilon r)$ is contained in a level i cluster C_i in the level-i partition $\Pi_i^{(j)}$; since a ball is a connected component, all shortest paths from s to vertices within $\mathbf{B}(s,\varepsilon r)$ must be contained within C_i in the level-i partition

$\Pi_i^{(j)}$.

3.2 Routing Table Construction

In this section, we focus on the process of building up routing tables once the nodes in the network have been assigned addresses that reflect their positions in each of the m decomposition trees. During this process, routing information is aggregated and exchanged between special nodes in different clusters at each level. We refer to such special nodes as exchange nodes (for routing) or entry points (for packet forwarding) of their corresponding clusters. The algorithm for selecting exchange nodes for each cluster is an independent issue that we do not address in this paper. Similar to the CER hierarchical routing scheme described in KK, no routing information describing the internal behavior of a cluster is propagated outside a cluster; hence a cluster is regarded from outside as a single node whose distance to itself is zero.

We use a modified version of the distributed Bellman-Ford algorithm as in Fig 3.2.2 to perform routing updates: especially, to establish the next-hop entries and update estimated path lengths for destination clusters in the routing tables for the basic model. For routing updates, we are going to focus on entries for one specific decomposition tree T_j that corresponds to $\mathbf{\Pi}^{(j)} = (\Pi_i^{(j)})_{i=0}^h$.

Let s and t be two neighboring nodes (that they are connected by a channel (s,t)) which belong to the same k^{th} level cluster $C_k \in \Pi_k^{(j)}$ and not to any lower level cluster in T_j, where $k \in \{1,2,\ldots,h\}$. Let $C_{k-1}(s), C_{k-1}(t) \in \Pi_{k-1}^{(j)}$ respectively denote the $k-1^{st}$ level clusters to which s and t each belong in tree T_j. Let $C_k(s,t)$ denote the level-k cluster that contains both s and t; note that $C_{k-1}(s), C_{k-1}(t) \subseteq C_k(s,t)$ in T_j since T_j represents a laminar decomposition. We use $\mathsf{lca}^j(s,t)$ to denote the lowest common ancestor of s and t in a particular tree T_j; hence $\mathsf{lca}^j(s,t) = C_k(s,t) \in \Pi_k^{(j)}$. For a pair of nodes s, t, $\mathsf{lca}^j(s,t)$ can be determined by inspecting the common prefixes of local addresses, $\mathsf{addr}_j(s)$ and $\mathsf{addr}_j(t)$.

Recall that in node s, for any cluster $C_i(s)$ in T_j that contains s at level i, for all $i = 0,\ldots,h$, routing table entries are kept for all clusters that are descendants of $C_i(s) \in \Pi_i^{(j)}$ within ℓ levels down a decomposition tree for $T_j, \forall j$. Thus each entry in the routing table Route_s for T_j corresponds to some level-(i') cluster $C' \in \Pi_{i'}^{(j)}$ in T_j, where $i' = 0, 1, \ldots, h-1$; that entry is also denoted as C' and indexed by the global address $\mathsf{addr}(C')$ of its associated cluster C', and contains the following fields in Route_s: **(a)** a next hop $\mathsf{NextHop}_s(\mathsf{addr}(C'))$ to reach C' from s, **(b)** a path length field $\mathsf{HF}(s, C')$ that is the current path length at node s for reaching cluster C' through $\mathsf{NextHop}_s(\mathsf{addr}(C'))$, and **(c)** a $\mathsf{ValidPath}_s(C')$ bit. Initially, the path length fields for all the entries in Route_s for tree T_j are set to ∞ except for the self entries as shown in the Initialization Procedure in Fig 3.2.2.

We use $C_i(s, C') = C_i(s) \in \Pi_i^{(j)}$ to denote the level-i common ancestor of s and $C' \in \Pi_{i'}^{(j)}$ such that $i \geq i' + 1$ and $C_i(s) \supseteq C'$. Note that $C_h(s, C') = C_h(s)$ contains $C' \in \Pi_{i'}^{(j)}$, for all $i' \leq h - 1$, since $C_h(s)$ contains the entire network. Similarly, we use $\mathsf{lca}^j(s, C')$ to denote the lowest common ancestor of s and $C' \in \Pi_{i'}^{(j)}$ in tree T_j, where $C' \subseteq \mathsf{lca}^j(s, C') \subseteq C_i(s, C')$ for all i such that $C' \subseteq C_i(s)$. For node s and cluster C', the $\mathsf{lca}^j(s, C')$ can be determined by inspecting the common prefixes of local addresses $\mathsf{addr}_j(s)$ and $\mathsf{addr}_j(C')$.

As a consequence of the routing table specification, routing table entries at node s and t at all levels below $k - \ell$ in T_j refer to different cluster destinations; whereas all the other entries from level $k - \ell$ up to h refer to the same cluster destinations in T_j. The objective of the updating procedure is to compare the estimated lengths of the paths from s or t to any common destination and to update the routing tables to reflect the shorter paths. Whenever s receives a route update from t, for each common destination cluster C', its corresponding entry is potentially updated with a new next hop $\mathsf{NextHop}_s(\mathsf{addr}(C'))$, the path length $\mathsf{HF}(s, C')$ through the new $\mathsf{NextHop}_s(\mathsf{addr}(C'))$ as in Step 2-4, and the $\mathsf{ValidPath}_s(C')$ bit as in Step 5-9 of the Route Update Procedure in Fig 3.2.2.

We have a slightly different way of setting the $\mathsf{ValidPath}_s(C')$ bit from that specified in Section 2.4.2 to maximize the chance of setting it `true`. However, as before, once the $\mathsf{ValidPath}_s(C')$ bit is set to be `true`, a shortest path from s to C' is indeed guaranteed by following the next hop in Route_s for an entry C' and that in Route_v of each subsequent nodes v along the path from s to an entry point of C'.

Let a common destination entry for T_j in Route_s and Route_t correspond to a level-(i') cluster $C' \in \Pi_{i'}^{(j)}$, where $i' \geq k - \ell$. We denote the level of $\mathsf{lca}^j(s, C')$ in T_j as l_0; The following inequalities, $i' + 1 \leq l_0 \leq i' + \ell$, must be satisfied for C' to be an entry in Route_s. The $\mathsf{ValidPath}_s(C')$ bit is set to be `true` so long as for "any" of the common ancestor $C_i(s,C')$ of s and C' at level i, for all $l_0 \leq i \leq i' + \ell$, both $\mathsf{HF}(s,C') \leq \varepsilon\rho^i$ and $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s,C')$ are true. It is set to be `false` otherwise. Note that when $i' \geq h - \ell$, both $\mathsf{HF}(s,C') \leq \Delta$ and $\mathbf{B}(s,\Delta) \subseteq C_h(s,C')$ are always true since $C_h(s,C')$ is the entire network; hence we set $\mathsf{ValidPath}_s(C')$ bit `true` for all C' at level $h - \ell$ and above in Step 5 of the Initialization Procedure.

The reason we set $\mathsf{ValidPath}_s(C')$ bit this way is the following. Recall that by constructing the m decomposition trees, each node s "knows" if $\mathbf{B}(s,\varepsilon\rho^i)$ is contained $C_i(s) \in \Pi_i^{(j)}$ in tree T_j; naturally, if $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s) \in \Pi_i^{(j)}$, then $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_l(s) \in \Pi_l^{(j)}$ is true for all $l \geq i$. However, if $\mathbf{B}(s,\varepsilon\rho^i) \not\subseteq C_i(s)$, we do not assume that we know information such as "whether a ball $\mathbf{B}(s,r)$ of a radius $\varepsilon\rho^i > r > \varepsilon\rho^{(i-1)}$ is contained in $C_i(s)$ or not", since that is not the type of information that our constructive algorithm provides by default; note that if $r \leq \varepsilon\rho^{(i-1)}$, we will just check if $\mathbf{B}(s,\varepsilon\rho^{(i-1)}) \subseteq C_{i-1}(s)$ to decide if $\mathbf{B}(s,r) \subseteq C_i(s)$. Our routing algorithm thus makes minimal assumptions about the information that is available at each node about balls around it being contained at a certain level or not.

Another specification in terms of routing that is different from that of Section 2.4.2 is the following. Assume we route a packet from s toward C'. Instead of assuming the packet should always enter a cluster C' through the closest point $x = \mathsf{close}_s(C')$ in C' to s, we only require that the packet enters C' through a closest *entry* point $e_0 \in C'$. Correspondingly, for node s and a level-(i') cluster $C' \in \Pi_{i'}^{(j)}$ in T_j, the function $\mathsf{NextHop}_s(\mathsf{addr}(C'))$ returns the next hop on the path from s to e_0 provided that the next hop does not leave the cluster C at level $(i' + \ell)$ that contains C', and null otherwise. Recall an entry point $e_0 \in C'$ advertises routes for C' it belongs to. Note also e_0 does not need to be the closest one to s in C' in order to achieve $(1+\tau)$-stretch routing. (This is also true for overlay routing.) As a basic routing scheme, we keep a next hop $\mathsf{NextHop}_s(\mathsf{addr}(C'))$ in Route_s toward a closest entry point $e_0 \in C'$ for the sake of routing table consistency that we will elaborate shortly.

For overlay routing, we keep the IP address of an arbitrary entry point e_0 to C' (instead of a next hop $\mathsf{NextHop}_s(\mathsf{addr}(C'))$ toward e_0), since IP routing will deliver a packet from s to e_0 directly given the IP address of e_0 without having to rely on hop-by-hop forwarding as in the basic model that we focus in this section.

Initialization Procedure: initialize Route_s for tree T_j at node s

1. For $i = 0, 1, \ldots, h$
2. $\quad$ $\mathsf{HF}(s, C_i(s)) = 0$, and $\mathsf{ValidPath}_s(C_i(s)) = \texttt{true}$
3. For all other entries $C' \not\ni s$, let i' = level of C' in tree T_j
4. $\quad$ $\mathsf{HF}(s, C') = \infty$
5. $\quad$ If $i' \geq h - \ell$, then $\mathsf{ValidPath}_s(C') = \texttt{true}$
6. End

Route Update Procedure: upon receiving a route update from t such that $\mathsf{lca}^j(s,t) = C_k$

1. For each common entry $C' \in \Pi_{i'}^{(j)}$, which represents a level-(i') cluster in T_j, where $i' \geq k - \ell$
2. $\quad$ If $\mathsf{HF}(s, C') > d(x,t) + \mathsf{HF}(t, C')$, then
3. $\quad\quad$ $\mathsf{HF}(s, C') \leftarrow d(x,t) + \mathsf{HF}(t, C')$
4. $\quad\quad$ nexthop field of $C' \leftarrow t$
5. $\quad\quad$ If $i' < h - \ell$, then
6. $\quad\quad\quad$ Let l_0 = level of $\mathsf{lca}^j(s, C')$ in T_j and m satisfies $\varepsilon\rho^{m-1} \leq \mathsf{HF}(s, C') \leq \varepsilon\rho^m$
7. $\quad\quad\quad$ for all levels $i : \max\{l_0, m\} \leq i \leq i' + \ell$
8. $\quad\quad\quad\quad$ If $\mathbf{B}(s, \varepsilon\rho^m) \subseteq \mathbf{B}(s, \varepsilon\rho^i) \subseteq C_i(s)$ in T_j, then
9. $\quad\quad\quad\quad\quad$ $\mathsf{ValidPath}_s(C') = \texttt{true}$
10. $\quad\quad\quad\quad\quad$ Goto 1
11. End

Figure 3.2.2. DISTRIBUTED BELLMAN-FORD Algorithm for T_j at Node s

Definition 3.1. *We call a path an internal path in cluster C if all the nodes in that path belong to C.*

Similar to KK, we define the *equilibrium* condition as the situation when no changes occur in the topology of network and the contents of $\mathsf{HF}(s, C')$ in the routing table reach "minimal" constant values after a certain number of updates.

Claim 3.1. *The distributed Bellman-Ford algorithm guarantees that in equilibrium condition, $\mathsf{HF}(s, C')$ will be the length of the shortest path from s to a closest entry point e_0 of C' when* $\mathsf{ValidPath}_s(C')$ *is* $\texttt{true}$*, i.e.,* $\mathsf{HF}(s, C') = d(s, e_0)$ *in* Route_s.

Proof. Let the level of $C' \in \Pi_{i'}^{(j)}$ in tree T_j be $i' < h - \ell$ and let the level of $\mathsf{lca}^j(s, C')$ be l_0. We only set $\mathsf{ValidPath}_s(C')$ $\texttt{true}$ in the routing algorithm when for "any" of the level-i cluster $C_i(s) \in \Pi_i^{(j)}$, where $l_0 \leq i \leq i' + \ell$, both

$\mathsf{HF}(s,C') \leq \varepsilon\rho^i$ and $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s) \in \Pi_i^{(j)}$ hold. Denote the lowest such level r, where $r \in [l_0, i'+\ell]$. All shortest paths from s to some entry point $x' \in C'$ of distance $d(s,x') \leq \mathsf{HF}(s,C') \leq \varepsilon\rho^r$ are thus internal to $C_r(s) \in \Pi_r^{(j)}$ in T_j, since such paths are contained in $\mathbf{B}(s,\varepsilon\rho^r)$, which is a connected component entirely contained in $C_r(s) \in \Pi_i^{(j)}$. Note that some $x' \in C'$ must have advertised itself as an entry point to C' for such paths to be established within $\{C_r(s) - C'\}$ and for $C' \in \Pi_{i'}^{(j)}$ to appear in Route_s. Thus $C' \subseteq C_r(s)$ since $x' \in \{C' \cap C_r(s)\} \neq \emptyset$ and $r > l_0$; we thus denote $C_r(s)$ as $C_r(s,C')$ from this point on.

In addition, every node $v \in C_r(s,C')$, including those along the shortest paths from s to x' inside $\mathbf{B}(s,\varepsilon\rho^r)$, contains a routing table entry to C', since it is a descendant of $C_r(s,C')$ within ℓ levels down the decomposition tree T_j. Propagation and subsequent updating of routing information among nodes of $C_r(s,C')$ is equivalent to finding minimum path internal to $C_r(s,C')$ from any node $v \in \{C_r(s,C') - C'\}$ to an entry point of C' that is closest to node v; for s, the closest entry point to C' is e_0.

Improvements are made sequentially at each update over the distance $\mathsf{HF}(u,C')$ from u to C' among nodes within $\mathbf{B}(s,\varepsilon\rho^r)$, until it reaches a minimal constant value if no changes occur in the topology of the network; hence all $u \in \mathbf{B}(s,\varepsilon\rho^r)$ "knows" how to route to C' with a path of bounded length. Given multiple entry points to C', the distributed Bellman-Ford algorithm guarantees that we find a shortest path not only to some entry point x' of C', but also to the closest, e_0 of C', from s in equilibrium condition, i.e., $\mathsf{HF}(s,C') = d(s,e_0)$. The entire path stays within $\mathbf{B}(s,\varepsilon\rho^r) \subseteq C_r(s,C')$, where r is specified as above.

Note that when $i' \geq h-\ell$, both $\mathsf{HF}(s,C') \leq \Delta$ and $\mathbf{B}(s,\Delta) \subseteq C_h(s,C')$ are always true since $C_h(s,C')$ is the entire network; hence we set $\mathsf{ValidPath}_s(C')$ `true` for all C' at level $h-\ell$ and above. The same argument as above applies to this case. □

The reason we require a closest entry point to C' is primarily for route convergence purpose when our protocol serves as an underlying routing scheme. For overlay routing, we allow an entry point to be any exchange node or simply a random node within the cluster, which is commonly assumed in peer-to-peer networks. Note that an exchange node of a given cluster is a node of that cluster which is connected to one or more nodes external to that cluster as defined in KK. We will use exchange node and entry point interchangeably unless we specify otherwise. The $(1+\tau)$-stretch property we are going to prove for hierarchical routing paths does not require the entry point for a cluster C' to be the closest to s either – a point that we will not elaborate on from now on.

Fact 3.1. *If a shortest path from s to e_0, an entry point to a level-(i') cluster $C' \in \Pi_{i'}^{(j)}$, is internal to $C_i(s) \in \Pi_i^{(j)}$ in tree T_j, where $i > i'$, then cluster $C' \ni e_0$ must be a sub-cluster that is entirely contained in $C_i(s)$ in T_j, i.e., $C' \subseteq C_i(s)$.*

Proof. First observe $e_0 \in \{C_i(s) \cap C'\}$, since shortest path from s to e_0 is internal to $C_i(s) \in \Pi_i^{(j)}$ in T_j. Since T_j represents a laminar decomposition, where a lower level cluster is always entirely contained in a higher level cluster, $e_0 \in C_i$ is sufficient to guarantee that $\{C' \ni t\} \subseteq C_i$. □

3.3 Path Characteristics

We forward packets according to the Forwarding algorithm in Figure 2.4.2. Let s and t be two arbitrary nodes. For destination t, let $C'(t)$ be the cluster whose $\mathsf{addr}(C'(t))$ is returned by the function $\mathsf{PrefMatch}_s(t)$ at Step 3 of the Forwarding algorithm. We assume, w.l.o.g., $C'(t) \in \Pi_{i'}^{(j)}$, i.e., $C'(t)$ is in the level-(i') partition $\Pi_{i'}^{(j)}$, where $i' \leq h - \ell$, in tree T_j. Recall that $h = \left\lceil \log_\rho \Delta \right\rceil$. Let $l_0 \leq h$ be the level of $\mathsf{lca}^j(s, C'(t)) \in \Pi_{l_0}^{(j)}$ in T_j.

We say $C'(t) \ni t$ is the cluster that has the longest *valid* prefix matching with t in Route_s, since the level of $C'(t)$ is the lowest across all trees among clusters C' in Route_s such that $C' \ni t$ and $\mathsf{ValidPath}_s(C')$ is `true`. Before we proceed, we first give more definitions, some of which are adapted from KK.

h_{st}^c: Length of the estimated minimum path from node s to node t as derived from the routing information at node s. (The superscript c stands for clustered routing.)

Exchange node e_z: a node of a cluster C that is connected to one or more nodes external to C.

$A_i(t)$: Subset of all exchange nodes (entry points) that connect a level-i cluster $C_i(t) \in \Pi_i^{(j)}$ in tree T_j, for all $j = 1, \ldots, m$, with any other level-i cluster within the same ancestor $C_n(t) \in \Pi_n^{(j)}$ in the same tree T_j, for all $n \leq i + \ell$. From the above definitions, all entry points e_z of $C'(t) \in \Pi_{i'}^{(j)}$ that connect $C'(t)$ to any other level-(i') cluster that stays within $C_{i'+\ell}(t) \in \Pi_{i'+\ell}^{(j)}$ in tree T_j hence belong to $A_{i'}(t)$.

Let $e_0 \in A_{i'}(t) \cap C'(t)$ be the closest entry point for s to reach $C'(t) \in \Pi_{i'}^{(j)}$ in T_j.

$\hat{C}_k(s,t)$: For $k \leq h-1$, $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ is the level-k cluster in T_j, where $l_0 \leq k \leq i' + \ell$, that is the lowest-level common cluster of s and t such that $\mathbf{B}(s,\varepsilon\rho^k) \subseteq \hat{C}_k(s,t)$ and $\mathbf{B}(s,\varepsilon\rho^k)$ contains a shortest path from s to $C'(t) \in \Pi_{i'}^{(j)}$ in T_j, where $i' \leq h-\ell$; such $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ always exists since we know $\mathbf{B}(s,\varepsilon\rho^r) \subseteq C_r(s,t)$ and $\mathsf{HF}(s,C'(t)) \leq \varepsilon\rho^r$ must both hold for some $l_0 \leq r \leq i' + \ell$, given that $\mathsf{ValidPath}_s(C'(t))$ is `true` in Route_s, due to the specification of the distributed Bellman-Ford algorithm. Let k be the lowest such level r. Note that $C'(t) \subseteq \hat{C}_k(s,t)$ since T_j represents a laminar decomposition and k is at least l_0. For $k = h$, $\hat{C}_k(s,t) = C_h(s,t) \in \Pi_h^{(j)}$, is the root cluster X of T_j that corresponds to the entire network G. In this case, $\hat{C}_k(s,t) = C_h(s,t)$ always contains all shortest paths from s to $C'(t) \in \Pi_{i'}^{(j)}$ in T_j, where $i' = h-\ell$, given that G is a connected graph.

$h_{se_z}^i(t)$: Length of the shortest path from node s to an exchange node $e_z \in A_{i'}(t) \cap C'(t)$ as contained in $\hat{C}_k(s,t)$ defined above. The superscript i stands for an internal path within $\hat{C}_k(s,t)$. At equilibrium, $h_{se_0}^i(t) = \mathsf{HF}(s,C'(t)) = d(s,e_0)$ since the shortest path from s to e_0 is internal to $\hat{C}_k(s,t)$, and by Claim 3.1, $\mathsf{HF}(s,C'(t)) = d(s,e_0)$ in Route_s given that $\mathsf{ValidPath}_s(C'(t))$ is `true` in Route_s and e_0 is the closest entry point to $C'(t)$ for node s. Recall that $\mathsf{HF}(s,C'(t))$ is the current path length filed in Route_s for node s to reach $C'(t) \in \Pi_{i'}^{(j)}$ via its current $\mathsf{NextHop}_s(\mathsf{addr}(C'(t)))$. Note when the shortest path from s to e_z is not internal to $\hat{C}_k(s,t)$, we denote it with $h_{se_z}^i = \infty$.

In order to reach t, function $\mathsf{PrefMatch}_s(t)$ is called by the Forwarding algorithm at node s, which looks across Route_s for all trees and picks a tree T_j that contains $C'(t)$ with a closest entry point $e_0 \in A_{i'}(t) \cap C'(t)$. Node s then stores $\langle \mathsf{addr}(t), \mathsf{addr}(C'(t)) \rangle$ in the packet header and sends the packet to $\mathsf{NextHop}_s(\mathsf{addr}(C'(t)))$; the packet header remains the same while intermediate nodes v forward the packet along a shortest path from s to e_0, that is contained in the common cluster $\hat{C}_k(s,t)$ of s and t in T_j, until it reaches e_0.

The key observation we have regarding a path h_{st}^c from s to t is the following. The path may not be contained within the lowest common ancestor $\mathsf{lca}^j(s,t) \in \Pi_{l_0}^{(j)}$

of s and t in a particular tree T_j. However, the segment from s to $C'(t)$, is contained within $\hat{C}_k(s,t)$ in T_j, where $l_0 \le k \le i' + \ell$, when following a shortest path from s to e_0, which is the closest entry point to $C'(t)$. Recall $\hat{C}_k(s,t)$ is a common cluster of s and $C'(t)$ at a level higher than that of $\mathsf{lca}^j(s,t)$. Conceptually, we route packets from s to t within $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ to avoid being stuck in $\mathsf{lca}^j(s,t) \in \Pi_{l_0}^{(j)}$, which may not contain any path (e.g., when $\mathsf{lca}^j(s,t)$ in T_j is disconnected) or contains only very long paths from s to t. The shortest path from s to e_0 is thus an internal path relative to $\hat{C}_k(s,t)$, which we denote with $h^i_{se_0}$.

Finally, We define a constant $\phi = \frac{4}{\rho^{\ell}\varepsilon}$ that we will use throughout this section. It is easy to verify that $2\eta_{i-\ell} \le \frac{2}{\rho^{\ell-1}(\rho-1)\varepsilon}\varepsilon\rho^i < \phi\varepsilon\rho^i$. Recall that $\ell = \Theta(\log_\rho 1/\varepsilon\tau)$ and $\rho = \Theta(\frac{1}{\varepsilon})$, where we choose suitable constants so that $\rho^2 \le \frac{1-\phi}{\phi}$ is satisfied. The rest of this section is dedicated to the proof of the main theorem of this section, before which we first prove two lemmas regarding the level of $C'(t)$ and $\hat{C}_k(s,t)$ given $d(s,t)$. Note that we always have $k \le h$ and $i' \le h - \ell$. We will ignore the case when $k = h$ until the end of this section.

Lemma 3.1. *Let $d(s,t) \le (1-\phi)\varepsilon\rho^i$, where $1 \le i \le h$. The cluster $C'(t) \in \Pi_{i'}^{(j)}$ in T_j that has the longest valid prefix matching with t with $\mathsf{ValidPath}_s(C'(t)) = \mathtt{true}$, is at a level $i' \le \max(0, i-\ell)$; the common cluster $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ as defined above that contains the shortest path from s to $C'(t)$ is at level $k \le i$.*

Proof. We first prove the lemma when $i \le \ell$ with the following claim.

Case $i \le \ell$.

Claim 3.2. *Let $\varepsilon\rho^{i-1} < d(s,t) \le \varepsilon\rho^i$ for $1 \le i \le \ell$. Then $C'(t) = C_0(t)$ is t itself; the lowest common cluster $\hat{C}_k(s,t)$ such that $\mathbf{B}(s,\varepsilon\rho^k) \subseteq \hat{C}_k(s,t)$ and $\mathbf{B}(s,\varepsilon\rho^k)$ contains the shortest path from s to $C_0(t)$, i.e., t itself, is at level $k = i$.*

Proof. Node s has a routing table entry for all t such that $d(s,t) \le \varepsilon\rho^{\ell}$, since $\mathbf{B}(s,d(s,t)) \subseteq \mathbf{B}(s,\varepsilon\rho^{\ell})$ is fully contained in some level-ℓ cluster $C_\ell(s) \in \Pi_\ell^{(q)}$ in some tree T_j, and $C'(t)$ is $C_0(t) \in \Pi_0^{(q)}$.

The properties of the (ρ,ε)-PPHD ensure that there is at least one tree T_j such that $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s) \in \Pi_i^{(j)}$ in T_j. Since $d(s,t) \le \varepsilon\rho^i$, we know that $t \in \mathbf{B}(s,\varepsilon\rho^i)$ and $C_0(t) \subseteq C_i(s)$ in T_j. The lowest common cluster $\hat{C}_k(s,t)$ such that $\mathbf{B}(s,\varepsilon\rho^k) \subseteq$

$\hat{C}_k(s,t)$ and $\mathbf{B}(s,\varepsilon\rho^k)$ contains the shortest path from s to $C'(t) = C_0(t)$, i.e., t itself, is $C_i(s) \in \Pi_i^{(j)}$ in tree T_j and $k = i$. □

We now prove the general case when $i > \ell$.

Case $h-1 \geq i > \ell$. Let $x' \in A_{i-\ell}(t)$ be an arbitrary entry point to some level-$(i-\ell)$ cluster $C \ni t$ in some tree; hence $d(x',t) \leq 2\eta_{i-\ell} \leq \phi\varepsilon\rho^i$ since $x',t \in C$. Applying the triangle inequality, we have $d(s,x') \leq d(s,t) + d(t,x') \leq \varepsilon\rho^i$; thus all shortest paths from s to x', for all $x' \in A_{i-\ell}(t)$, are contained in $\mathbf{B}(s,\varepsilon\rho^i)$.

The properties of the (ρ,ε)-PPHD ensure that there is at least one tree T_q such that $\mathbf{B}(s,\varepsilon\rho^i)$ is not cut in the level-i partition $\Pi_i^{(q)}$; let $C_i(s) \in \Pi_i^{(q)}$ be the level-i cluster in T_q such that $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s)$. Since $d(s,t) \leq (1-\phi)\varepsilon\rho^i$, we have $t \in \mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s) \in \Pi_i^{(q)}$. Let $C_{i-\ell}(t) \in \Pi_{i-\ell}^{(q)}$ be the level-$(i-\ell)$ cluster in T_q containing t; we know that $C_{i-\ell}(t) \subseteq C_i(s)$, since $t \in \{C_{i-\ell}(t) \cap C_i(s)\}$ and T_q represents a laminar decomposition. Hence we have $C_i(s) = C_i(t) = C_i(s,t)$ in the level-i partition $\Pi_i^{(q)}$ in tree T_q.

The $\mathsf{ValidPath}_s(C_{i-\ell}(t))$ bit must be set `true` in Route_s by the distributed Bellman-Ford algorithm in node s, since (a) $\mathbf{B}(s,\varepsilon\rho^i) \subseteq C_i(s,t) \in \Pi_i^{(q)}$, and (b) $\mathsf{HF}(s,C_{i-\ell}(t)) \leq \varepsilon\rho^i$ in Route_s for entry $C_{i-\ell}(t) \in \Pi_{i-\ell}^{(q)}$ in tree T_q at equilibrium, given that all shortest paths from s to an entry point x', for all $x' \in A_{i-\ell}(t) \cap C_{i-\ell}(t)$, are internal to $\mathbf{B}(s,\varepsilon\rho^i)$. Thus $\mathsf{PrefMatch}_s(t)$ can (and may indeed) just return $\mathsf{addr}(C_{i-\ell}(t))$ given no "better" choices, in which case, $i' = i-\ell$ and $k \leq i$.

However, $\mathsf{PrefMatch}_s(t)$ always finds a cluster $C'(t) \in \Pi_{i'}^{(j)}$ at the *lowest* level across all trees, such that $t \in C'(t)$ and $\mathsf{ValidPath}_s(C'(t))$ is `true` in Route_s; hence $C'(t)$ is at level $i' \leq i-\ell$.

We know that $\mathbf{B}(s,\varepsilon\rho^r) \subseteq C_r(s,t) \in \Pi_r^{(j)}$ and $\mathsf{HF}(s,C'(t)) \leq \varepsilon\rho^r$ must both hold, for some $l_0 \leq r \leq i' + \ell$, in order for $\mathsf{ValidPath}_s(C'(t))$ bit to be `true`, due to the specification of the distributed Bellman-Ford algorithm. Let k be the lowest such r; we have $k \leq i' + \ell \leq i$ for $i > \ell$.

Case $i = h$. We have $k \leq h$ and $i' \leq h - \ell$ trivially, since both holds for all possible distances of $d(s,t)$ up to Δ, which is the diameter of the network G. □

Claim 3.3. *When $C'(t) \in \Pi_1^{(j)}$ is at level 1, $d(s,t) > \varepsilon\rho^\ell$.*

Proof. Prove by contradiction. Assume that $d(s,t) \leq \varepsilon\rho^{\ell}$. By Claim 3.2, we have $C'(t) = C_0(t)$, contradicting the assumption that $C'(t)$ is at level 1. □

Lemma 3.2. *Let $C'(t) \in \Pi_{i'}^{(j)}$ be the cluster returned by function $\mathsf{PrefMatch}_s(t)$ at Step 3 in the Forwarding algorithm and its level be i', where $h - \ell \geq i' \geq 1$. Then $d(s,t) > (1-\phi)\varepsilon\rho^{i'+\ell-1}$.*

Proof. Prove by contradiction. Assume that $d(s,t) \leq (1-\phi)\varepsilon\rho^{i'+\ell-1}$. By lemma 3.1, the cluster $C'(t) \ni t$ that has the longest valid prefix matching with t with the $\mathsf{ValidPath}_s(C'(t))$ bit set `true` in Route_s is at level at most $i'-1$, thus contradicting the assumption that $C'(t)$ is at level i'. □

We next prove the following lemma regarding the level of $C'(t)$ given the level of $\hat{C}_k(s,t)$.

Lemma 3.3. *Let a level-k cluster $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ in tree T_j, where $h-1 \geq k \geq \ell$, be the lowest-level common cluster of s and t such that a shortest path from s to $C'(t) \in \Pi_{i'}^{(j)}$ is contained in $\mathbf{B}(s,\varepsilon\rho^k) \subseteq \hat{C}_k(s,t) \in \Pi_k^{(j)}$. Then $C'(t) \in \Pi_{i'}^{(j)}$ is at either level $k-\ell$ or level $k-\ell+1$.*

Proof. Be definition of $\hat{C}_k(s,t)$, we know that $l_0 \leq k \leq i' + \ell$ and $l_0 \geq i'+1$, where l_0 is the level of $\mathsf{lca}^j(s, C'(t))$. Thus $k - \ell \leq i' \leq k-1$. The lowest level that $C'(t)$ can be is at $k-\ell$, and we argue that $C'(t)$ can not be at a level higher than $k-\ell+1$.

Let e_0 be a closest entry point to $C'(t)$ for s, such that $e_0 \in A_{i'}(t) \cap C'(t)$ and the shortest path from s to e_0 is internal to $\mathbf{B}(s,\varepsilon\rho^k) \subseteq \hat{C}_k(s,t) \in \Pi_k^{(j)}$; hence $d(s,e_0) \leq \varepsilon\rho^k$. Since $C'(t)$ is at least one level below $\hat{C}_k(s,t)$ in T_j and $e_0, t \in C'(t)$, we have $d(e_0,t) \leq 2\eta_{k-1}$. Note that $C'(t) \subseteq \hat{C}_k(s,t)$ by Fact 3.1. Applying the triangle inequality, we have $d(s,t) \leq d(s,e_0) + d(e_0,t) \leq \varepsilon\rho^k + 2\eta_{k-1}$.

Now we examine the distance of $d(s,y')$ for all $y' \in A_{k-\ell+1}(t)$. Given that $d(t,y') \leq 2\eta_{k-\ell+1}$, we apply the triangle inequality and obtain:

$$\begin{aligned} d(s,y') &\leq d(s,t) + d(t,y') \\ &\leq d(s,e_0) + d(e_0,t) + d(t,y') \\ &\leq \varepsilon\rho^k + 2\eta_{k-1} + 2\eta_{k-\ell+1} \\ &\leq \varepsilon\rho^{k+1}, \end{aligned}$$

where $\ell \geq 2$ and $\rho = \Theta(\frac{1}{\varepsilon})$.

Thus all shortest paths from s to y', for all $y' \in A_{k-\ell+1}(t)$, are contained in $\mathbf{B}(s, \varepsilon\rho^{k+1})$. The properties of the (ρ,ε)-PPHD ensure that there is at least one tree $T_{j'}$ such that $\mathbf{B}(s, \varepsilon\rho^{(k+1)}) \subseteq C_{k+1}(s) \in \Pi_{k+1}^{(j')}$. Let $C_{k-\ell+1}(t) \in \Pi_{k-\ell+1}^{(j')}$ be the level-$(k-\ell+1)$ cluster that contains t in $T_{j'}$. Given that $t \in \mathbf{B}(s, \varepsilon\rho^{k+1}) \subseteq C_{k+1}(s)$, we know that $C_{k-\ell+1}(t) \subseteq C_{k+1}(s) \in \Pi_{k+1}^{(j')}$ since $t \in \{C_{k-\ell+1}(t) \cap C_{k+1}(s)\} \neq \emptyset$ and $T_{j'}$ represents a laminar decomposition. Thus $C_{k-\ell+1}(t) \in \Pi_{k-\ell+1}^{(j')}$ must appear in s' routing table with $\mathsf{ValidPath}_s(C_{k-\ell+1}(t))$ set `true`, since $C_{k-\ell+1}(t) \subseteq C_{k+1}(s)$ is within ℓ levels below $C_{k+1}(s)$ in $T_{j'}$ and all shortest paths from s to $C_{k-\ell+1}(t)$ are contained in $\mathbf{B}(s, \varepsilon\rho^{k+1}) \subseteq C_{k+1}(s)$ in $T_{j'}$.

Thus the level i' of $C'(t)$ must satisfy $k-\ell \leq i' \leq k-\ell+1$ for $C'(t) \in \Pi_{i'}^{(j)}$ to be returned by $\mathsf{PrefMatch}_s(t)$. □

Fact 3.2. *When $k = h$ and $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ is $C_h(s,t) \in \Pi_h^{(j)}$, which is the entire network G, we know that $C'(t) \in \Pi_{i'}^{(j)}$ is at level $i' = h-\ell = k-\ell$.*

The next lemma shows the path characteristics from s to t up till entry point e_0 of $C'(t) \in \Pi_{i'}^{(j)}$.

Lemma 3.4. *All messages to be forwarded or sent from node s to node t will follow the same shortest path up to the closest entry point e_0 of $C'(t) \in \Pi_{i'}^{(j)}$ to s. The shortest path from s to e_0 is internal to $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ in T_j; it has a length of $h_{se_0}^i$ that satisfies:*

$$h_{se_0}^i = min_{e_z \in A_{i'}(t) \cap C'(t)} \{h_{se_z}^i\}, \tag{3.3.1}$$

where i' is the level of $C'(t) \in \Pi_{i'}^{(j)}$ and $k \leq i'+\ell$, and $\hat{C}_k(s,t) \in \Pi_k^{(j)}$, $A_{i'}(t)$, and $h_{se_z}^i$ are as defined above, and $h_{se_z}^i = \infty$ when the shortest path from s to e_z is not contained in $\hat{C}_k(s,t)$. At equilibrium, $h_{se_0}^i = \mathit{HF}(s, C'(t)) = d(s,e_0)$. Finally, all vertices v on the shortest path from s to e_0 have a non-null $\mathit{NextHop}_v(\mathit{addr}(C'(t)))$ and share the same closest entry point e_0 to cluster $C'(t)$.

Proof. By the definition of $\hat{C}_k(s,t)$, for $k \leq h-1$, we know that $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ is the level-k cluster, where $l_0 \leq k \leq i'+\ell$, in tree T_j, that is the lowest-level common cluster of s and t such that $\mathbf{B}(s, \varepsilon\rho^k) \subseteq \hat{C}_k(s,t) \in \Pi_k^{(j)}$ and $\mathbf{B}(s, \varepsilon\rho^k)$ contains a shortest path from s to $C'(t) \in \Pi_{i'}^{(j)}$ in T_j; specifically, $\mathbf{B}(s, \varepsilon\rho^k)$ contains a shortest paths from s to e_0, and $d(s, e_0) \leq \varepsilon\rho^k$. By Fact 3.1, we have $C'(t) \subseteq \hat{C}_k(s,t)$. When $k = h$, $\hat{C}_k(s,t) = C_h(s,t) \in \Pi_h^{(j)}$ is the root cluster X and naturally contains all shortest paths from s to $C'(t) \subseteq C_h(s,t)$, given that G is a connected graph.

All the nodes in $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ contain one entry for $C'(t) \in \Pi_{i'}^{(j)}$ in their routing tables, since $k \leq i' + \ell$ and $C'(t) \subseteq \hat{C}_k(s,t)$ is a cluster within ℓ levels below $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ in tree T_j. Propagation and subsequent updating of routing information among nodes of $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ in T_j is equivalent to finding the minimum path internal to $\hat{C}_k(s,t)$ from any node $u \in \{\hat{C}_k(s,t) - C'(t)\}$ to an entry point of $C'(t)$ that is closest to node u; for s, the closest entry point to $C'(t)$ is e_0 such that $h_{se_0}^i = min_{e_z \in A_{i'}(t) \cap C'(t)}\{h_{se_z}^i\}$. Hence, at equilibrium, e_0 is on the minimal path from s to $C'(t)$ and $h_{se_0}^i = \mathsf{HF}(s, C'(t))$ represents the length of such minimal path.

All shortest paths of length $d(s, e_0)$ from s to e_0 are internal to $\hat{C}_k(s,t)$; when $k \leq h-1$, it is within $\mathbf{B}(s, \varepsilon\rho^k) \subseteq \hat{C}_k(s,t) \in \Pi_k^{(j)}$. Hence, at equilibrium, within $\mathbf{B}(s, \varepsilon\rho^k) \in \hat{C}_k(s,t) \in \Pi_k^{(j)}$ for $k \leq h-1$, or within $\hat{C}_k(s,t) = X$ for $k = h$, a shortest path of length $h_{se_0}^i = d(s, e_0)$ is formed between s and e_0 among nodes within a connected component, that share a common entry for $C'(t) \subseteq \hat{C}_k(s,t)$ in their routing tables. Thus we have $\mathsf{HF}(s, C'(t)) = h_{se_0}^i = d(s, e_0)$.

For any node v on one of these shortest paths from s to e_0, s and v must share the same closest entry point e_0 to $C'(t)$ at equilibrium, due to the execution of the distributed Bellman-Ford algorithm; furthermore, intermediate nodes v will be able to route the packet toward $C'(t) \in \Pi_{i'}^{(j)}$ in $\hat{C}_k(s,t)$ consistently since they each contain an entry for $C'(t) \in \Pi_{i'}^{(j)}$ with a non-null $\mathsf{NextHop}_v(\mathsf{addr}(C'(t)))$ field, given that these paths stay within $\hat{C}_k(s,t) \in \Pi_k^{(j)}$, where $k \leq i' + \ell$. The Forwarding algorithm will forward messages from node s destined to node t along the shortest path thus formed to first reach $C'(t)$ in tree T_j. $\square$

The process of finding the next entry point repeats by the time the packet reaches e_0, an entry point to $C'(t) \in \Pi_{i'}^{(j)}$ in tree T_j, until the packet reaches its destination t. For example, e_0 selects a new tree T_l that contains the next cluster $C''(t) \in \Pi_{i''}^{(l)}$ with a longer prefix matching with t than $C'(t)$, and updates the packet header with $C''(t)$ accordingly. Note that $C''(t)$ and $C'(t)$ may belong to two different trees; hence while intermediate nodes between one entry point and another never switch trees, upon reaching an entry point, it is free to switch. The next lemma states the upper bound on the level i'' of $C''(t) \in \Pi_{i''}^{(l)}$, and the level of the common cluster $\hat{C}_k(x,t) \in \Pi_k^{(l)}$ that contains a shortest path from x to $C''(t)$ in $\mathbf{B}(x, \varepsilon\rho^k) \in \hat{C}_k(x,t) \in \Pi_k^{(l)}$. If x is t itself, we are done with forwarding.

Lemma 3.5. *Let $1 \le i' \le h - \ell$ be the level of $C'(t) \in \Pi_{i'}^{(j)}$. Once the packet from s reaches an entry point x in $A_{i'}(t) \cap C'(t)$, including e_0, x will find a new level-(i'') cluster $C''(t) \in \Pi_{i''}^{(l)}$ at level $i'' \le \max(0, i'-2)$ in some tree T_l, and the common cluster $\hat{C}_k(x,t) \in \Pi_k^{(l)}$ as defined above is at a level $k \le i' - 2 + \ell$.*

Proof. We have $d(x,t) \le 2\eta_{i'} \le \phi\varepsilon\rho^{i'+\ell}$, since $x \in A_{i'}(t) \cap C'(t)$ is an entry point to some level-(i') cluster $C'(t) \in \Pi_{i'}^{(j)}$ containing t. We have $d(x,t) \le (1-\phi)\varepsilon\rho^{i'-2+\ell}$ so long as $\rho^2 \le \frac{1-\phi}{\phi}$, which can be satisfied when suitable constants are chosen for $\ell = \Theta(\log_\rho 1/\varepsilon\tau)$ and $\rho = \Theta(\frac{1}{\varepsilon})$. Lemma 3.1 tells us that $k' \le i' - 2 + \ell$ and $C''(t) \in \Pi_{i''}^{(l)})$ is at level $\le \max(0, i'-2)$. □

We are now ready for the main theorem that summarizes the path properties.

Theorem 3.1. *Follow the Forwarding algorithm in Section 2.4.3, for all $k \le h$, the path from s to t as derived from the routing information at node s satisfies the recursive equation below, $h_{st}^c = h_{se_0}^i + h_{e_0t}^c$, where the shortest path $h_{se_0}^i$ from s to e_0 is contained in $\hat{C}_k(s,t)$ and its properties are as specified in Lemma 3.4. Secondly, the lookup path has a stretch of at most $(1+\tau)$. Finally, the algorithm switches trees for at most $\max(0, k-\ell+1)$ times. When $d(s,t) \le (1-\phi)\varepsilon\rho^n$, where $n \le h$, we have $k \le n$; otherwise, $k \le h$.*

Proof. The proof of the theorem is by induction on k, which is the level of the lowest common cluster $\hat{C}_k(s,t)$ of s and $C'(t)$ such that a shortest path from s to $C'(t)$ is contained in (a) $\mathbf{B}(s, \varepsilon\rho^k) \subseteq \hat{C}_k(s,t)$ for $k \le h-1$, or in (b) $\hat{C}_k(s,t) = C_h(s,t)$ for $k = h$. Recall i' is the level of $C'(t) \in \Pi_{i'}^{(j)}$, and $e_0 \in A_{i'}(t) \cap C'(t)$ is the closest entry point to $C'(t)$ for node s within $\hat{C}_k(s,t) \in \Pi_k^{(j)}$.

Base Case: $k \le \ell - 1$**.**

We first prove the following claim.

Claim 3.4. *If $\hat{C}_k(s,t)$ is at level $k \le \ell - 1$, then $C'(t) = C_0(t)$ and $d(s,t) \le \varepsilon\rho^{\ell-1}$.*

Proof. By the definition of $\hat{C}_k(s,t)$, we know $\mathbf{B}(s, \varepsilon\rho^k) \subseteq \hat{C}_k(s,t) \in \Pi_k^{(j)}$ in T_j and $d(s,e_0) \le \varepsilon\rho^k$, where $e_0 \in C'(t)$ is the closest entry point to $C'(t)$ for node s. Thus $d(s,e_0) \le \varepsilon\rho^{\ell-1}$ for $k \le \ell - 1$. Since $C'(t) \in \Pi_{i'}^{(j)}$ is a descendant of $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ in T_j, it must be at a level lower than k; hence $d(e_0,t) \le 2\eta_{\ell-2} \le \frac{2\rho^{\ell-1}}{\rho-1} \le 4\rho^{\ell-2}$, since $e_0, t \in C'(t)$, and $C'(t)$ is at level $i' \le \ell - 2$.

Applying the triangle inequality, we have $d(s,t) \le d(s,e_0) + d(e_0,t) \le \varepsilon\rho^{\ell-1} + 4\rho^{\ell-2} \le \varepsilon\rho^{\ell}$. Thus by Claim 3.2, we have $C'(t) = C_0(t)$, which is t itself; furthermore, $e_0 = t$ and $d(s,t) = d(s,e_0) \le \varepsilon\rho^{\ell-1}$. □

The above claim shows that $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ in tree T_j contains a shortest path from s to $C'(t) = C_0(t) \in \Pi_0^{(j)}$, and t is the closest entry point to $C_0(t)$, which is t itself. Thus $h^c_{e_0t} = h^c_{tt} = 0$, since a node's distance to itself is zero. It remains to show that $h^c_{st} = h^i_{se_0} = h^i_{st}$; recall h^i_{st} refers to the shortest path from s to t as included in $\hat{C}_k(s,t)$. This is true since the routing table of every node v in $\hat{C}_k(s,t) \in \Pi_k^{(j)}$ for $k \le \ell - 1$ contains an entry for $C_0(t) = t$, and a shortest path from s to t is contained in $\mathbf{B}(s, \varepsilon\rho^k) \subseteq \hat{C}_k(s,t)$ in tree T_j; hence at equilibrium, the clustered path between s and t as derived from Route_s is the shortest path from s to t, and it is internal to $\hat{C}_k(s,t)$, i.e., $h^c_{st} = \mathsf{HF}(s, C_0(t)) = h^i_{st} = d(s,t)$, where $k \le \ell - 1$. The stretch is exactly 1 since $\frac{h^c_{st}}{d(s,t)} = \frac{h^i_{st}}{d(s,t)} = 1$. The forwarding algorithm does not switch tree at all.

Case $k = \ell$**.** By Lemma 3.3, $C'(t)$ is at level 0 or 1. When $C'(t)$ is at level $k - \ell = 0$, the proof is the same as that in the base case.

When $C'(t)$ is at level $k - \ell + 1 = 1$, we have $d(s,t) > \varepsilon\rho^{\ell}$ by Claim 3.3. All messages to be forwarded or sent from node s to node t will first follow the same shortest path of length $h^i_{se_0} = d(s,e_0)$, that is internal to $\hat{C}_\ell(s,t)$, up to the closest entry point e_0 of $C'(t)$, as specified in Lemma 3.4.

Upon reaching e_0, Lemma 3.5 can be applied to show that $h^c_{e_0t}$, the clustered path from e_0 to t, is entirely contained in a level-(k') cluster $\hat{C}_{k'}(e_0,t)$ in some tree $T_{j'}$, where $k' \le i' - 2 + \ell = \ell - 1$; thus as proved in the base case, $h^c_{e_0t} = h^i_{e_0t} = d(e_0,t)$. The clustered path from s to t as derived from Route_s indeed satisfies $h^c_{st} = h^i_{se_0} + h^c_{e_0t}$, where $h^i_{se_0} = d(s,e_0)$ and $h^c_{e_0t} = d(e_0,t)$.

Hence we obtain the bound on the entire path: $h^c_{st} = d(s,e_0) + d(e_0,t) \le d(s,t) + 2d(e_0,t)$, where $d(s,e_0) \le d(s,t) + d(e_0,t)$ by triangle inequality. And the path stretch is: $\frac{h^c_{st}}{d(s,t)} = 1 + \frac{2d(e_0,t)}{d(s,t)} \leq 1 + \frac{2(\rho+1)}{\varepsilon\rho^{\ell}} \leq 1+\tau$, where $\ell \geq 1 + (\log_\rho 4/\varepsilon\tau)$. The algorithm switches trees at most once.

Case $k \geq \ell+1$**.** First we assume that the theorem is true up to $k-1$, let us show that it is true for k.

Let $\hat{C}_k(s,t) \in \Pi^{(j)}_k$ be the k^{th} level cluster that contains a shortest path from s to $C'(t) \in \Pi^{(j)}_{i'}$ in T_j. According to Lemma 3.4, all messages to be forwarded or sent from node s to node t will first follow the same shortest path of length $h^i_{se_0}$, that is internal to $\hat{C}_k(s,t)$, up to the closest entry point e_0 of $C'(t)$. By Lemma 3.3, we know that $C'(t)$ is at level $k-\ell$ or $k-\ell+1$ when $k \leq h-1$. When $k=h$, $C'(t)$ is at level $h-\ell = k-\ell$.

Upon reaching e_0, Step 3 of the forwarding algorithm is applied to find $C''(t)$, that has the longest valid matching with t in e_0's routing table. Since $C'(t)$ is at level $k-\ell$ or $k-\ell+1$, Lemma 3.5 shows the lowest common cluster $\hat{C}_{k'}(e_0,t)$ of e_0 and t, such that $\mathbf{B}(e_0,\varepsilon\rho^{k'}) \subseteq \hat{C}_{k'}(e_0,t)$ and $\mathbf{B}(e_0,\varepsilon\rho^{k'})$ contains a shortest path from e_0 to $C''(t)$, is at level $k' \leq i'-2+\ell \leq k-1$. Thus $h^c_{e_0t}$ is known from the induction hypothesis and $h^c_{st} = h^i_{se_0} + h^c_{e_0t}$.

Now we proceed to prove the bound on the stretch for level k. Let $C'(t)$ be at level $\beta-\ell$, where β is k or $k+1$; hence $d(e_0,t) \leq 2\eta_{\beta-\ell}$ given that $e_0,t \in C'(t)$. By Lemma 3.2, $d(s,t) > (1-\phi)\varepsilon\rho^{\beta-1}$, where $i' = \beta-\ell \geq 1$ for all $k \geq \ell+1$.

By Lemma 3.4, $h^i_{se_0}$ is the shortest path from s to e_0 that is internal to $\hat{C}_k(s,t)$ and $h^i_{se_0} = d(s,e_0)$; applying the triangle inequality, we obtain:

$$h^i_{se_0} = d(s,e_0) \leq d(s,t) + d(e_0,t) \leq d(s,t) + 2\eta_{\beta-\ell}. \tag{3.3.2}$$

By the induction hypothesis, we know

$$h^c_{e_0t} \leq (1+\tau)d(e_0,t) \leq 2(1+\tau)\eta_{\beta-\ell}. \tag{3.3.3}$$

Finally, we get the bound on the total path length from s to t:

$$h^c_{st} = h^i_{se_0} + h^c_{e_0t} \leq d(s,t) + 2(2+\tau)\eta_{\beta-\ell} \tag{3.3.4}$$

Now using the fact that $d(s,t) \geq (1-\phi)\varepsilon\rho^{\beta-1}$, and the fact that $\tau \leq 1$, we obtain the path stretch from s to t:

$$\frac{h^c_{st}}{d(s,t)} = 1 + \frac{2(2+\tau)\eta_{\beta-\ell}}{d(s,t)} \leq 1 + \frac{6\eta_{\beta-\ell}}{(1-\phi)\varepsilon\rho^{\beta-1}} \leq 1+\tau, \tag{3.3.5}$$

where $\ell \geq (\log_\rho 8/\varepsilon\tau) + 2$.

Finally, the algorithm switches trees for at most $k-\ell$ times to finally route within a level ℓ cluster, after which it switches tree at most once, thus adding up to a total number of $k-\ell+1$ times.

Now we look at the bound on k itself. When $d(s,t) \leq (1-\phi)\varepsilon\rho^n$, for all $n \leq h$, we have $k \leq n$ by Lemma 3.1. We now verify that all statements in the theorem still apply, for the clustered path h^c_{st}, when $d(s,t) > (1-\phi)\varepsilon\rho^{(h)}$ and $\hat{C}_k(s,t)$ is $C_h(s,t)$. First of all, when $\hat{C}_k(s,t) = C_h(s,t)$, following the Forwarding algorithm in Section 2.4.3, we know that $C'(t)$ is at level $h-\ell$ and hence $d(s,t) > (1-\phi)\varepsilon\rho^{(h-1)}$ by Lemma 3.2. The shortest path from s to $e_0 \in C'(t)$, $h^i_{se_0}$, is internal to $C_h(s,t)$, the entire network G. Upon reaching e_0, $h^c_{e_0t}$ is known by applying the theorem directly since $d(e_0,t) \leq 2\eta_{h-\ell} \leq (1-\phi)\varepsilon\rho^{h-1}$. Thus we have $h^c_{e_0t} \leq (1+\tau)d(e_0,t) \leq (1+\tau)2\eta_{h-\ell}$. Hence, the entire path satisfies the equation, $h^c_{st} = h^i_{se_0} + h^c_{e_0t}$. Second, with the same calculation as the proof above, it is easy to verify that the entire path h^c_{st} has a stretch of at most $(1+\tau)$ given that $d(s,t) \geq (1-\phi)\varepsilon\rho^{(h-1)}$ and $h^c_{e_0t} \leq (1+\tau)d(e_0,t) \leq (1+\tau)2\eta_{h-\ell}$. The algorithm switches trees for at most $(h-\ell+1)$ times. □

Corollary 3.1. *For all t such that $d(s,t) \leq \varepsilon\rho^\ell$, path stretch is* 1.

Proof. Node s has a routing table entry for all t such that $d(s,t) \leq \varepsilon\rho^\ell$, since $\mathbf{B}(s,d(s,t)) \subseteq \mathbf{B}(s,\varepsilon\rho^\ell)$ is fully contained in some level-ℓ cluster $C_\ell(s) \in \Pi^{(j)}_\ell$ in some tree T_j, and $C'(t)$ is $C_0(t) \in \Pi^{(j)}_0$; the base case of the above proof shows that path stretch is 1. □

Part II: Edge Disjoint Paths

4 Edge-disjoint Paths in Moderately Connected Graphs

In the next three chapters, we prove the following theorem regarding undirected EDP.

Theorem 4.1. *There is a* polylog n*-approximation algorithm for the edge disjoint path problem in a general graph* $\mathcal{G}$ *with minimum cut and node degree* $\Omega(\log^5 n)$.

4.1 The Approach

We begin with a fractional relaxation of the problem, where each terminal pair can route a real-valued amount of flow between 0 and 1, and this flow can be split fractionally across a set of distinct paths. This can be expressed as an LP and can be solved efficiently. We denote the value of an optimal fractional LP solution as OPT^*. Our algorithm routes a polylogarithmic fraction of this value using integral edge-disjoint paths.

The algorithm proceeds by decomposing the graph into well-connected subgraphs, based on OPT^*, so that a subset of the terminal pairs, that remain within each subgraph are "well-connected", following a decomposition procedure of from Chekuri et al. [2005]. Then, for each well connected subgraph G, we construct an expander graph that can be embedded into G using its terminal set. We use a result by Khandekar, Rao and Vazirani in Khandekar et al. [2006], where they show that one can build an expander graph H on a set of nodes V by constructing $O(\log^2 n)$ perfect matchings $M_1, \ldots, M_{O(\log^2 n)}$ between $O(\log^2 n)$ sets of equal partitions of V in an iterative manner.

Our contribution along this line is to route each perfect matching $M_t, \forall t$, on one of the $O(\log^2 n)$ (edge-disjoint) subgraphs of G. The "splitting procedure", motivated by Karger's theorem Karger [1994], simply assigns edges of G uniformly at random into $O(\log^2 n)$ subgraphs. Using Karger's arguments, we show that all cuts in each subgraph have approximately the correct size with high probability. Here we crucially use the polylogarithmic lower bound on the min-cut. We then route each matching M_t on a unique split subgraph using a max-flow computation with unit capacities. Thus, we can route all $O(\log^2 n)$ matchings edge disjointly in G and embed an expander graph H integrally with congestion 1 on G.

After we construct such an expander graph H for each G, we route terminal pairs in H greedily via short paths. This is effective since there are plenty of short disjoint paths in an expander graph Broder et al. [1994]; Kleinberg and Rubinfeld [1996]. Since a node in H maps to a cluster of nodes in G that is connected by a spanning tree, we put a capacity constraint on $V(H)$: we allow only a single path to go through each node. We greedily connect a pair of terminals from G via a path in H while taking both nodes and edges along the chosen path away from H, until no short paths remain between any unrouted terminal pair. For the pairs we indeed route, we know the congestion is 1 in the original graph G, since we use each edge and node in H only once, and edges and nodes of H correspond to disjoint paths of G. We use a lemma in Garg et al. [1993] to show that such a greedy method ensures that we route a sufficiently large number of such pairs; We note that this method was proposed but analyzed somewhat differently by Kleinberg and Rubinfeld [1996]. Our analysis is more like that of Obata [2004], and yields somewhat stronger bounds. Our approximation factor is $O(\log^{10} n)$. (A breakdown of this factor is described in Theorem 6.2.)

4.1.1 Related Work

Much of recent work on EDP has focused on understanding the polynomial-time approximability of the problem. Previously, constant or polylogarithmic approximation algorithms were known for trees with parallel edges Garg et al. [1993], expanders Kleinberg and Rubinfeld [1996]; Kolman and Scheideler [2001], grids and grid-like graphs Aumann and Rabani [1995]; Awerbuch et al. [1994]; Kleinberg and Tardos [1995a,b], and even-degree planar graphs Kleinberg [2005]. For general graphs, the best approximation ratio for EDP in directed graphs is $O(\min(n^{2/3}, \sqrt{m}))$ Chekuri and Khanna [2003]; Kleinberg [1996]; Kolliopoulos and Stein [1998]; Srinivasan [1997]; Varadarajan and Venkataraman [2004], where m denotes number of edges in the input graph. This is matched by the $\Omega(m^{\frac{1}{2}-\varepsilon})$-hardness of approximation result by Guruswami et al Guruswami et al. [1999]. For undirected and directed acyclic graphs, the upper bound has been improved to $O(\sqrt{n})$ Chekuri et al. [2006b]. For even-degree planar graphs, an $O(\log^2 n)$-approximation Kleinberg [2005] is obtained recently.

A variant is the EDP with Congestion (EDPwC) problem, where the goal is to route as many terminals as possible, such that at most ω demands can go through any edge in the graph. For EDPwC on planar graphs, for $\omega = 2$ and 4, $O(\log n)$ Chekuri et al. [2004b, 2005] and constant Chekuri et al. [2006a] approximations have been obtained respectively. For undirected graphs, the hardness results Andrews et al. [2005] are $\Omega(\log^{1/2-\varepsilon} n)$ for EDP and $\Omega(\log^{(1-\varepsilon)/(\omega+1)} n)$ for EDPwC.

A closely related problem is the congestion minimization problem: Given a graph and a set of terminal pairs, connect *all* pairs with integral paths while minimizing the maximum number of paths through any edge. Raghavan and Thompson [1987] show that by applying a randomized rounding to a linear relaxation of the problem one obtains an $O(\log n / \log\log n)$ approximation for both directed and undirected graphs. For hardness of approximation, Andrews and Zhang [2005a] show a result of $\Omega((\log\log^{1-\varepsilon} m))$ for undirected and an almost-tight result Andrews and Zhang [2006] of $\Omega(\log^{1-\varepsilon} m)$ for directed graphs, improving that of $\Omega(\log\log m)$ by Chuzhoy and Naor Chuzhoy and Naor [2004]. Finally, the All-or-Nothing Flow (ANF) problem Chekuri et al. [2004a, 2005] is to choose a subset of terminal pairs such that for each chosen pair, one can fractionally route a unit of flow for all the chosen pairs. The hardness result for ANF and ANF with Congestion is the same as that of EDP and EDPwC Andrews et al. [2005]. Currently, there exists an $O(\log^2 n)$ Chekuri et al. [2005] approximation for ANF. Indeed, we build on the techniques developed in this approximation algorithm for ANF. This ratio directly contributes to our approximation factor.

We summarize these results in Table 4.1.1.

	directed?	Hardness of Approx.	Upperbound
MinCong	no	$\Omega(\frac{\log\log n}{\log\log\log n})$ Andrews and Zhang [2005a]	$O(\log n/\log\log n)$ Raghavan and Thompson [1987]
MinCong	yes	$\Omega((\log\log^{1-\varepsilon} m)$ Andrews and Zhang [2006]	$O(\log n/\log\log n)$ Raghavan and Thompson [1987]
EDP	yes	$\Omega(m^{\frac{1}{2}-\varepsilon})$ Guruswami et al. [1999]	$O(\min(n^{2/3}, \sqrt{m}))$ Chekuri and Khanna [2003]; Kleinberg [1996] Kolliopoulos and Stein [1998]; Srinivasan [1997] Varadarajan and Venkataraman [2004]
EDP	no	$\Omega(\log^{\frac{1}{2}-\varepsilon} n)$ Andrews et al. [2005]	$O(\sqrt{n})$ Chekuri et al. [2006b]
EDPwC	no	$\Omega(\log^{\frac{1-\varepsilon}{\omega+1}} n)$ for $w = o\left(\frac{\log\log n}{\log\log\log n}\right)$ Andrews et al. [2005]	
EDPwC	no	superconstant for $w = \frac{\eta\log\log n}{\log\log\log n}$ Andrews et al. [2005]	
ANFwC	no	$\Omega(\log^{\frac{1-\varepsilon}{\omega+1}} n)$ for $w = o(\frac{\log\log n}{\log\log\log n})$ Andrews et al. [2005]	$O(\log^2 n)$ for $\omega = 1$ Chekuri et al. [2005]

Table 4.1. HARDNESS OF APPROXIMATIONS AND UPPER BOUNDS.

4.2 Definitions and Preliminaries

We work with graph $G = (V,E)$ with unit-capacity edges, where we allow parallel edges, unless we specify a capacity function for edges explicitly. For a capacitated graph $G = (V,E,c)$, where c is an integer capacity function on edges, one can replace each edge $e \in E$ with $c(e)$ parallel edges. For a cut $(S, \bar{S} = V \setminus S)$ in G, let $\delta_G(S)$, or simply $\delta(S)$ when it is clear, denote the set of edges with exactly one endpoint in S in G. Let $\mathsf{cap}(S,\bar{S}) = |\delta_G(S)|$ denote the total capacity of edges in the cut. The edge expansion of a cut $(S,\bar{S})$, where $|S| \leq |V|/2$, is $\phi(S) = \frac{\mathsf{cap}(S,\bar{S})}{|S|}$. The expansion of a graph G is the minimum expansion over all cuts in G. We call a graph G an expander if its expansion is at least a constant.

An instance of a routing problem consists of a graph $\mathcal{G} = (V,E)$ and a set of terminals pairs $\mathcal{T} = \{(s_1,t_1),(s_2,t_2),\ldots,(s_k,t_k)\}$. Nodes in $\mathcal{T}$ are referred to as terminals. Given an EDP instance $(\mathcal{G},\mathcal{T})$ with k pairs of terminals, we will use the following LP relaxation as specified in (4.2.1), to obtain an optimal fractional solution. Let $\mathcal{P}_i, \forall i$, denote the set of paths joining s_i and t_i in $\mathcal{G}$.

$$\begin{aligned} \max \sum_{i=1}^{k} x_i \quad s.t. & & & (4.2.1)\\ x_i - \sum_{p \in \mathcal{P}_i} f(p) &= 0, \forall 1 \leq i \leq k & & (4.2.2)\\ \sum_{p: e \in p} f(p) &\leq 1, \forall e \in E & & (4.2.3)\\ x_i, f(p) &\in [0,1], \forall 1 \leq i \leq k, \forall p & & (4.2.4) \end{aligned}$$

We let $\mathsf{OPT}^*(\mathcal{G},\mathcal{T})$ be the value of this linear program for the optimal solution $\bar{f}$ of the LP. In the text, where we always refer to a single instance, we primarily use OPT^*.

Given a non-negative weight function $\vec{\pi} : X \to \mathbb{R}^+$ on a set of nodes X in G, we use following definitions from Chekuri et al. [2005].

Definition 4.1. (CKS2005 Chekuri et al. [2005]) *X is $\vec{\pi}$-cut-linked in G if $\forall S$ such that $\vec{\pi}(S \cap X) = \sum_{x \in S \cap X} \vec{\pi}(x) \leq \vec{\pi}(X)/2$, $|\delta(S)| \geq \vec{\pi}(S \cap X)$; We also refer to (G,X) as a $\vec{\pi}$-cut-linked instance.*

Definition 4.2. (CKS2005 Chekuri et al. [2005]) *A set X is $\vec{\pi}$-flow-linked in G if there is a feasible multicommodity flow for the problem with demand $\mathsf{dem}(u,v) = \vec{\pi}(u)\vec{\pi}(v)/\vec{\pi}(X)$ between every unordered pair of terminals $u,v \in X$.*

Remark 4.1. *Note this is a product flow with $\mathsf{dem}(u,v) = w(u)w(v)$, where $w(u) = \vec{\pi}(u)/\sqrt{\vec{\pi}(X)}$.*

We have the following proposition immediately from the definitions above.

Proposition 4.1. (CKS2005 Chekuri et al. [2005]) *If a set X is $\vec{\pi}$-flow-linked in G, then it is $\vec{\pi}/2$-cut-linked. If X is $\vec{\pi}$-cut-linked in G, then it is $\vec{\pi}/\beta(G)$-flow-linked, where $\beta(G)$ is the worst-case mincut-maxflow gap on product multicommodity flow instances on $\mathcal{G}$.*

Definition 4.3. (CKS2005 Chekuri et al. [2005]) *A set of nodes X is well-linked in G if $\forall S$ such that $|S \cap X| \leq |X|/2$, $|\delta(S)| \geq |S \cap X|$.*

4.3 Decomposition of the Input Instance

In this section, we first present Theorem 4.2 regarding a preprocessing phase of our algorithm that decomposes and processes $(\mathcal{G},\mathcal{T})$ into a collection of cut-linked instances with a min-cut $\Omega(\log^3 n)$ in each subgraph. We then state our main theorem with a breakdown of the polylog n approximation factor. Finally, we give an outline on how we route terminal pairs in each cut-linked instance (G,T); Note that we use G to refer to a subgraph that we obtain through Theorem 4.2 starting from Section 6.1 till the end of the paper, while $\mathcal{G}$ refers to the original input graph. We first specify the following parameters.

- **Parameters related to original EDP instance** $(\mathcal{G},\mathcal{T})$
  - $\omega \log^2 n$ is the number of matchings as in Figure 6.1.1;
  - min-cut $\kappa = \Omega(\log^3 n) = \frac{12(\ln n)(\omega \log^2 n+1)}{\varepsilon^2}$, where $\varepsilon < 1$;
  - $\beta(\mathcal{G}) = O(\log n)$: as in Proposition 4.1 for $\mathcal{G}$.
  - $\lambda(n) = 10\beta(\mathcal{G}) \log \mathsf{OPT}^*(\mathcal{G},\mathcal{T}) = O(\log^2 n)$: as introduced in Theorem 5.1 in Chekuri et al. [2005].

Theorem 4.2. *There is a polynomial time decomposition algorithm, that given an EDP instance* $(\mathcal{G}, \mathcal{T})$*, where* $\mathcal{G}$ *has a min-cut of size* $\Omega(\kappa \log^2 n)$*, and a solution* $\bar{f}$ *to the fractional EDP problem, with* $x_i, \forall i$*, being specified as in (4.2.1), produces a disjoint set of subgraphs and a weight function* $\vec{\pi} : V(\mathcal{G}) \to \mathbb{R}^+$ *on* $V(\mathcal{G})$ *where*

(1) *there are* $\alpha_1, \ldots, \alpha_k$ *such that* $\forall u$ *in a subgraph* H*,* $\vec{\pi}(u) = \sum_{i:s_i=u,t_i \in H} \alpha_i x_i$*, (note that this implies* $\forall s_i t_i \in \mathcal{T}$*,* x_i *contributes the same amount of weight to* $\vec{\pi}(s_i)$ *and* $\vec{\pi}(t_i)$*);*

(2) *the set of nodes* $V(H)$ *in each subgraph* H *is* $\vec{\pi}$*-cut-linked in* H*;*

(3) *each subgraph* H *has min-cut* $\kappa = \Omega(\log^3 n)$*;*

(4) $\forall u$ *in a subgraph* H *s.t.* $\vec{\pi}(H) \geq \Omega(\log^3 n)$*,* $\vec{\pi}(u) \leq \sum_{i:s_i=u,t_i \in H} \frac{x_i}{\beta(\mathcal{G})\lambda(n)}$*;*

(5) *and* $\vec{\pi}(\mathcal{G}) = \Omega(\mathit{OPT}^* / \beta(\mathcal{G})\lambda(n))$*.*

The decomposition essentially says that summing across all subgraphs G, a fair fraction of terminal pairs in $\mathcal{T}$ remain (condition 4, 5); indeed, we lose only a constant fraction of the terminal pairs (by assigning a zero weight to those lost terminals) of $\mathcal{T}$. In addition, each subgraph G is well connected with respect to X, the set of induced terminals of $\mathcal{T}$ in G, in the sense of (G, X) being a $\vec{\pi}$-cut-linked instance. This decomposition is essentially the same as Theorem 5.1 of Chekuri, Khanna, and Shepherd Chekuri et al. [2005]. We need to do some additional work to ensure that the min-cut condition (condition 3) holds. We prove a flow-based version of the result in Section 5.1.

In particular, we sketch a proof to Theorem 5.2, which states a more refined and stronger version of Theorem 4.2. Actual proof of Theorem 5.2 is shown in Section 5.3.

5 Obtaining a Cut-Linked Decomposition

5.1 An Outline of the Decomposition Procedure

In this section, we first sketch a proof to Theorem 5.2, which states a more refined and stronger version of Theorem 4.2. Actual proof of Theorem 5.2 is shown in Section 5.3.

We first transform $(\mathcal{G},\mathcal{T})$ to a set of flow-linked instances by following a decomposition procedure in CKS05 Chekuri et al. [2005], the outcome of which is summarized in the following theorem.

5.1.1 The CKS Flow-Linked Decomposition Theorem

Theorem 5.1. (CKS2005 Chekuri et al. [2005]) *Let $\mathsf{OPT}^*(\mathcal{G},\mathcal{T})$ be a solution to the LP for a given instance $(\mathcal{G},\mathcal{T})$ of EDP in an input graph $\mathcal{G}$. One can efficiently compute a partition of $\mathcal{G}$ into node-disjoint induced subgraphs G_1, G_2, $\ldots$, G_ℓ, and weight functions $\vec{\pi} : V(G_i) \to \mathbb{R}^+$ with the following properties. Let $\mathcal{T}_i$ be the induced pairs of $\mathcal{T}$ in G_i and let X_i be the set of terminals of $\mathcal{T}_i$.*

(1) $\vec{\pi}_i(u) = \vec{\pi}_i(v)$ *for* $uv \in \mathcal{T}_i$.

(2) X_i *is* $\vec{\pi}_i$*-flow-linked in* G_i.

(3) $\sum_{i=1}^{\ell} \vec{\pi}_i(X_i) = \Omega(\mathsf{OPT}^*(\mathcal{G},\mathcal{T})/\lambda(n))$, *where* $\lambda(n) = 10\beta(\mathcal{G}) \log \mathsf{OPT}^*(\mathcal{G},\mathcal{T})$.

Remark 5.1. *Although the statement of condition 1 in the CKS decomposition theorem assumes that each node u belongs to only a single terminal pair in $\mathcal{T}$, their actual proof does not depend on such an assumption.*

The proof of the theorem appears in CKS05 Chekuri et al. [2005]. They use this procedure as the first step in a two-step transformation from the optimal multicommodity flow solutions $\bar{f}$ to obtain sets of well-linked terminal sets, that eventually leads to an $O(\log^2 K)$-approximation for the ANF problem described in Section 4.1, where $K = |\mathcal{T}|$. We place details regarding this decomposition in the Section 5.3. From now on, we refer to both $(G_i, \mathcal{T}_i)$ and (G_i, X_i) as a $\vec{\pi}_i$-flow-linked instance without differentiation.

5.1.2 Processing Subgraphs to Maintain Mincut Condition

We treat the induced subproblems $(G_i, \mathcal{T}_i), \forall i$ independently. Given (G_i, X_i) such that X_i is $\vec{\pi}_i$-flow-linked in G_i, there are two post-processing stages.

(1) **Min-cut processing stage.** Formally, let $V(G_i)$ be the current set of vertices of G_i. We keep cutting off the smaller side S of a minimum cut, in terms of weight $\vec{\pi}_i$, from G_i when $\mathsf{cap}(S, V(G_i) \setminus S)$ is less than $\hat{c}$, until every cut in G_i is at least $\hat{c}$, where we set $\hat{c} = \Omega(\log^3 n)$.

    By cutting off, we remove both nodes in S and edges that are adjacent to S in current G_i; this includes the cases when we get rid of any single node whose degree fall below $\hat{c}$ from its original degree of $\Omega(\log^5 n)$. We call such a stage a min-cut processing stage.

(2) **Sparsest-cut processing stage.** In order to guarantee that we have an instance X_i' that is $\vec{\pi}_i'$-flow-linked in G_i for a new weight function $\vec{\pi}_i'$, we need to further "mute" some terminals with a positive weight under $\vec{\pi}$ by setting their weight to zero under $\vec{\pi}_i'$. This way, we can guarantee that every cut in G_i is good with respect to a product multicommodity flow demand that is defined based on the new weight function $\vec{\pi}_i'$. We emphasize that we do not remove any nodes or edges in this stage; hence the min-cuts are guaranteed to be $\Omega(\log^3 n)$.

5.1.3 A Modified Flow-Linked Decomposition Theorem

Therefore, we have the following theorem about the instances that we have by the end of this post-processing stage. The proof of this theorem is in Section 5.3.

Theorem 5.2. *Given a graph $\mathcal{G}$ with min-cut value $C^0 \geq (4a_0\lambda(n) + a_0 + 2)\hat{c}$, for some $a_0 \geq 2$. By the end of the sparsest-cut processing, we obtain a set of node-disjoint induced subgraphs $\hat{G}_1, \ldots, \hat{G}_\ell$, all with min-cut $\hat{c}$, and the corresponding disjoint subsets $\mathcal{T}'_1, \ldots, \mathcal{T}'_\ell$ of $\mathcal{T}$, such that terminals pairs in $\mathcal{T}'_i$ belong to $\hat{G}_i$ and there exist a set of weight functions $\vec{\pi}'_i : V(\hat{G}_i) \to \mathbb{R}^+$ with the following properties. Let X'_i be the set of terminals of T'_i.*

- (1) *there are $\alpha_1, \ldots, \alpha_k$ such that $\forall u$ in a subgraph $\hat{G}_i$, $\vec{\pi}(u) = \sum_{i:s_i=u, t_i\in\hat{G}_i} \alpha_i x_i$, (note that this implies $\forall s_i t_i \in \mathcal{T}'_i$, x_i contributes the same amount of weight to $\vec{\pi}'_i(s_i)$ and $\vec{\pi}'_i(t_i)$);*
- (2) *X'_i is $\vec{\pi}'_i$-flow-linked in G'_i;*
- (3) *$\forall u$ in a subgraph $\hat{G}_i$ s.t. $\vec{\pi}'_i(X'_i) \geq \Omega(\log^3 n)$, $\vec{\pi}'_i(u) \leq \sum_{i:s_i=u,t_i\in\hat{G}_i} \frac{x_i}{\beta(\mathcal{G})\lambda(n)}$;*
- (4) *$\sum_{i=1}^{\ell} \vec{\pi}'_i(X'_i) = \Omega(\mathit{OPT}^*(\mathcal{G},\mathcal{T})/\lambda(n)\beta(\mathcal{G}))$, where $\lambda(n) = \beta(\mathcal{G}) \log \mathit{OPT}^*(\mathcal{G}, T)$ and $\beta(G)$ is the worst-case mincut-maxflow gap on product multicommodity flow instances on $\mathcal{G}$.*

Finally, we define a weight function on $V(\mathcal{G})$ as follows: (a) $\forall i, \forall u \in \hat{G}_i$, where $\hat{G}_i$ is a subgraph of $\mathcal{G}$, we assign $\vec{\pi}(u) = \vec{\pi}'_i(u)/2$; and (b) assign $\vec{\pi}(u) = 0$, for nodes of $V(G)$ not in any $\hat{G}_i$. We thus have defined the weight function $\vec{\pi} : V(G) \to \mathbb{R}^+$ on the entire set of nodes of $\mathcal{G}$ as required by Theorem 4.2 with the same decomposition as we obtain for Theorem 5.2.

5.2 Details Regarding CKS Flow-linked Decompositions

The following notation appears in proof of Theorem 5.1 as in Chekuri et al. [2005]. We will inherit these in our proofs in Section 5.3. Let $H = (V(H), E(H))$ be a node induced subgraph of $G = (V, E)$.

- $\gamma(\mathcal{G}) = \mathsf{OPT}^*(\mathcal{G}, \mathcal{T})$.

- $\gamma(H) = \sum_{P\in\mathcal{P}:P\in H} \bar{f}(P)$: the total flow induced in H by the original flow $\bar{f}$; it counts flow only on flows paths $\bar{f}(P)$ from the the original flow path decomposition that are completely contained in H. $\mathcal{P}$ refers to the entire set of paths from the original flow decomposition.

- $\gamma(u,H)$: the flow in H for u, hence $\gamma(H) = 1/2\sum_{u\in V(H)} \gamma(u,H)$.

Recall the following results from their decomposition procedure. Let $G_1, G_2, \ldots, G_\ell$ be the subgraphs produced by the decomposition.

(1) If $\gamma(G_i) \leq \lambda(n)/10$, assign $\vec{\pi}_i(u) = \vec{\pi}_i(v) = 1$ for some pair $uv \in \mathcal{T}_i$ with positive flow in G_i; and $\vec{\pi}_i(y) = 0$ for $y \neq u, v$. Hence one can just route a unit flow between the chosen pair $uv \in \mathcal{T}_i$ along an integral path; such a path exists since G_i is a connected component.

(2) Else, for $\gamma(G_i) > \lambda(n)/10$, X_i is $\vec{\pi}_i$-flow-linked in G_i, where $\vec{\pi}_i$ is defined as follows for G_i; Recall $\lambda(n) = 10\beta(\mathcal{G})\log \mathsf{OPT}^*(\mathcal{G},\mathcal{T})$.

   (a) $\vec{\pi}_i(u) = \frac{\gamma(u,G_i)}{\lambda(n)}, \forall u \in X_i$

   (b) $\vec{\pi}_i(u) = \gamma(u,G_i) = 0$ for $u \notin X_i$

Remark 5.2. *For both cases, the CKS weight function on $V(G_i)$ satisfy $\vec{\pi}_i(G_i) = \Omega(\gamma(G_i)/\lambda(n))$, given that $\vec{\pi}_i(G_i) = \sum_{x\in X_i} \vec{\pi}_i(x) = \sum_{x\in V(G_i)} \gamma(x,G_i)/\lambda(n) = 2\gamma(G_i)/\lambda(n)$; And the flow that one route in G_i satisfies the following two equivalent conditions.*

(*1*) *Define $\forall uv \in V(G_i)$,*

$$\mathsf{dem}^{\omega}(u,v) = \frac{\gamma(u,G_i)\gamma(v,G_i)}{\gamma(G_i)}, \tag{5.2.1}$$

as demands for the multicommodity product flow problem based on original induced flow values at each node $u \in V(G_i)$ of $\bar{f}$ in G_i; in $G_i, \forall i$, the concurrent max-flow value f for product flow $\mathsf{dem}^{\omega}(u,v)$, satisfy

$$f \geq f_0 = \frac{1}{2\lambda(n)}. \tag{5.2.2}$$

Thus $f_0 \mathsf{dem}^{\omega}(u,v)$ *units of demands can be simultaneously routed* $\forall uv$ *in* G_i *with congestion* 1.

(2) *For a scaled-down product flow problem* $\mathsf{dem}^{\vec{\pi}_i}(u,v)$*, such that each demand is* f_0 *of the original,* $\forall uv \in V(G_i)$*,*

$$\mathsf{dem}^{\vec{\pi}_i}(u,v) = \frac{\vec{\pi}_i(u)\vec{\pi}_i(v)}{\vec{\pi}_i(X_i)} = \frac{\gamma(u,G_i)\gamma(v,G_i)}{2\lambda(n)\gamma(G_i)} = \frac{\mathsf{dem}^{\omega}(u,v)}{2\lambda(n)} = f_0 \mathsf{dem}^{\omega}(u,v), \tag{5.2.3}$$

there is is a feasible flow in G_i *since the concurrent max-flow value is at least* 1.

Depending on the context, we may prefer to use the original product flow $\mathsf{dem}^{\omega}(u,v)$ than the feasible product flow $\mathsf{dem}^{\vec{\pi}_i}(u,v)$, or the other way around.

5.3 An Analysis on Postprocessing to Maintain Cut Conditions

The analysis of this section will lead to the proof of Theorem 5.2 eventually. Throughout this section, we keep reducing the set of terminals pairs of $\mathcal{T}_i$ that are relevant, in the sense that these pairs will remain to be candidate pairs that we eventually route edge disjointly in $\mathcal{G}$. Therefore, we keep track of the following set of parameters in each subgraph G_i that we obtain through flow decomposition:

- $\mathcal{T}_i$: the induced pairs of $\mathcal{T}$ in G_i that we still consider to route edge disjointly.
- A weight function $\vec{\pi}_i$ defined on the $V(G_i)$, with positive values only on terminals X_i of $\mathcal{T}_i$.

Finally, we use remaining-flow to keep track of the total remaining flows of $\bar{f}$ between terminal pairs in $\mathcal{T}_i$, across all i; note that remaining-flow is the lower bound on $\sum_i |\mathcal{T}_i|$.

By the end of the CKS flow decomposition, $\mathcal{T}_i$ is the induced pairs of $\mathcal{T}$ in G_i. There exists at least one flow path between a pair of terminals $uv \in T_i$, with a positive amount of flow, from original flow path decomposition of $\bar{f}$ that is entirely contained in G_i. We lose at most half of $\bar{f}$, where $\left|\bar{f}\right| = \mathsf{OPT}^*(\mathcal{G},\mathcal{T})$, because the number of edges that were cut during flow decomposition is at most $\mathsf{OPT}^*/2 = \gamma(G)/2$; hence

$$\mathsf{remaining\text{-}flow} \geq \sum_{i=1}^{\ell} \gamma(G_i) \geq \mathsf{OPT}^*(\mathcal{G},\mathcal{T})/2 = \gamma(G)/2, \tag{5.3.4}$$

and the total amount of the weights across all clusters is at least:

$$\sum_{i=1}^{\ell} \vec{\pi}_i(X_i) = \sum_{i=1}^{\ell} 2\gamma(G_i)/\lambda(n) = \Omega(\mathsf{OPT}^*(\mathcal{G},\mathcal{T})/\lambda(n)). \tag{5.3.5}$$

We are going to keep computing the original flows of $\bar{f}$ that we lose during the post-processing stages.

We specify the following parameters that are related to minimum cuts:

(1) $\hat{c}$: the smallest minimum cut value that we allow in G_i, $\forall i$, which is $\theta(\log^3 n)$.

(2) $\mathcal{C}^0$: the minimum cut value in original graph $\mathcal{G}$, which is $\Omega(\log^5 n)$.

(3) $\ell(S) = \mathsf{cap}(S, V \setminus S)$: size of a cut $(S, V \setminus S)$ in original graph $G = (V, E)$.

(4) $\mathsf{LOSS} \leq \mathsf{OPT}^*(\mathcal{G},\mathcal{T})/2$: number of edges that are cut during the CKS flow-decomposition process.

We analyze the minimum cut processing stage in the next two sections. Formally, let $V(G_i)$ be the current set of vertices of G_i. We keep cutting off the smaller side S of a minimum cut, in terms of weight $\vec{\pi}_i$, from G_i when $\mathsf{cap}(S, V(G_i) \setminus S)$ is less than $\hat{c}$, until every cut in G_i is at least $\hat{c}$. By cutting off, we remove both nodes in S and edges that are adjacent to S in current G_i.

Let $S_i^1, S_i^2, \ldots, S_i^{x_i}$ be the sets of vertices that we take away from G_i and in that order. We define the following notation to track this process of updating G_i.

- $G_i^0 = (V_i^0, E_i^0)$: the subgraph G_i before any of $S_i^t, t = 1, \ldots, x_i$ have been take out.

- X_i^0: the set of terminals of G_i^0 right after flow decomposition, such that X_i^0 is $\vec{\pi}_i$-flow-linked in G_i^0 as guaranteed by CKS decomposition.

– $G_i^t = (V_i^t, E_i^t), \forall t = 1, \ldots, x_i$: the remaining subgraph of G_i^0 after removing $S_i^1, \ldots, S_i^t$ and their adjacent edges; hence $V_i^t = V_i^0 \setminus \cup_{j=1,\ldots,t} S_i^j$.
– $\hat{G}_i = (\hat{V}_i, \hat{E}_i) = G_i^{x_i} = (V_i^{x_i}, E_i^{x_i})$ be the remaining subgraph of G_i^0 by the end of the min-cut processing stage.

5.3.1 Bound Edges Lost Due to Min-Cut Processing

Denote the number of edges that we take away from G_i^0 due to the min-cut processing by $\mathsf{edge\text{-}loss}_i, \forall i$.

Definition 5.1. *$\mathsf{edge\text{-}loss}_i$ is the sum of capacities of the minimum cuts that have caused $S_i^1, \ldots, S_i^{x_i}$ to be cut off from $G_i, \forall i$. Denote the sum of $\mathsf{edge\text{-}loss}_i$ across all i with $\mathsf{edge\text{-}loss}$,*

$$\mathit{edge\text{-}loss} = \sum_{i=1,2,\ldots} \mathit{edge\text{-}loss}_i = \sum_{i=1,2,\ldots} \sum_{t=1,\ldots,x_i} \mathit{cap}(S_i^t, V_i^t).$$

Remark 5.3. *Note that the number of edges that we take away from the final set of nodes $V(G_i) = V_i^{x_i} = V_i^0 \setminus \cup_{j=1,\ldots,x_i} S_i^j$ during the min-cut processing stage is upper bounded, and in fact may be smaller than $\mathsf{edge\text{-}loss}_i, \forall i$.*

We prove the following lemma in this section.

Lemma 5.1. *The total number of edges that we take away from decomposed subgraphs $G_i^0, G_i^1, \ldots$ is at most*

$$\mathit{edge\text{-}loss} = \sum_{i=1,2,\ldots} \mathit{edge\text{-}loss}_i \leq \frac{2\mathit{LOSS} \cdot \hat{c}}{C^0 - 2\hat{c}}. \tag{5.3.6}$$

Proof. We use a potential function $\psi(G_i)$ to count the number of edges we lose from nodes currently in G_i, as compared to the original graph $G = (V, E)$, while G_i keeps shrinking due to its min-cut processing. The counting process is as follows. We start with a component G_i such that $\psi_i^0 = \mathsf{LOSS}_i$ denotes the number of edges that we initially lose from nodes in G_i^0 right after the CKS flow decomposition procedure. Hence

$$\psi_i^0 = \psi(G_i^0) = \mathsf{LOSS}_i \geq 0, \tag{5.3.7}$$

and

$$\sum_{i=1,2,\ldots} \mathsf{LOSS}_i = 2\mathsf{LOSS}. \tag{5.3.8}$$

When a subset S is cut off, it claims away some credit from the current $\psi(G_i)$, since S is cut off because $\mathsf{cap}(S, V \setminus S)$ has decreased from above $\mathcal{C}^0$ to its current size in G_i, $\mathsf{cap}(S, V(G_i) \setminus S) \leq \hat{c}$ due to edges lost from nodes in S during CKS flow decomposition. That is, the amount of edge loss from nodes in S has contributed to the current value of $\psi(G_i)$.

Let ψ_i^t be value of $\psi(G_i)$ after taking t sets of vertices $S_i^1, \ldots, S_i^t$ and their adjacent edges away from G_i. Let $(S_i^{t+1}, V_i^t \setminus S_i^{t+1})$ be the minimum cut in G_i^t, and hence S_i^{t+1} be the $(t+1)^{st}$ set of vertices that we cut off from G_i because $\mathsf{cap}(S_i^{t+1}, V_i^t \setminus S_i^{t+1})$ is less than $\hat{c}$. The amount of credit S_i^{t+1} takes away from $\psi(G_i)$ is $(\mathsf{cap}(S_i^{t+1}, V \setminus S_i^{t+1}) - \mathsf{cap}(S_i^{t+1}, V_i^t \setminus S_i^{t+1}))$ and the credit it puts back is $\mathsf{cap}(S_i^{t+1}, V_i^t \setminus S_i^{t+1})$, since we remove edges in $(S_i^{t+1}, V_i^t \setminus S_i^{t+1})$ from G_i^t, in addition to the subgraph induced by S_i^{t+1} in G_i^t.

Let us denote the size of the original cut $(S_i^{t+1}, V \setminus S_i^{t+1})$ in $\mathcal{G}$ with

$$\ell_i^{t+1} = \ell(S_i^{t+1}) = \mathsf{cap}(S_i^{t+1}, V \setminus S_i^{t+1}) \geq \mathcal{C}^0. \tag{5.3.9}$$

Hence, we update $\psi(G_i)$ as follows,

$$\begin{aligned} \psi_i^{t+1} &= \psi_i^t - (\mathsf{cap}(S_i^{t+1}, V \setminus S_i^{t+1}) - \mathsf{cap}(S_i^{t+1}, V_i^t \setminus S_i^{t+1})) + \mathsf{cap}(S_i^{t+1}, V_i^t \setminus S_i^{t+1}) \\ &= \psi_i^t - (\ell_i^{t+1} - \mathsf{cap}(S_i^{t+1}, V_i^{t+1})) + \mathsf{cap}(S_i^{t+1}, V_i^{t+1}). \end{aligned}$$

Since $\mathsf{cap}(S_i^{t+1}, V_i^{t+1}) \leq \hat{c}$, we have

$$\psi_i^{t+1} \leq \psi_i^t - (\ell_i^{t+1} - \hat{c}) + \hat{c}. \tag{5.3.10}$$

Since the credit that a cut puts back is much less than the credit that it spent, there is only finite number x_i of such small cuts in G_i, $\forall i$. By the end of x_i rounds, there must be a non-negative credit in $\psi(G_i)$, since nodes in current G_i can never gain any edges. Hence

$$0 \le \psi(G_i) = \psi_i^x \le \mathsf{LOSS}_i - (\ell_i^1 - \hat{c}) + \hat{c} - (\ell_i^2 - \hat{c}) + \hat{c} - \ldots - (\ell_i^{x_i} - \hat{c}) + \hat{c}.$$

Summing the above inequalities over all i,

$$\sum_{i=1,2,\ldots} x_i \cdot C^0 \le \sum_{i=1,2,\ldots} \sum_{j=1,2,\ldots,x_i} \ell_i^j \le 2 \cdot \mathsf{LOSS} + 2 \sum_{i=1,2,\ldots} x_i \cdot \hat{c}.$$

Hence the total number of minimum cuts across all G_i that we process is

$$\sum_{i=1,2,\ldots} x_i \le \frac{2\mathsf{LOSS}}{C^0 - 2\hat{c}}. \tag{5.3.11}$$

Denote the sum of $\mathsf{edge\text{-}loss}_i$ across all i with $\mathsf{edge\text{-}loss}$ and thus

$$\begin{aligned}
\mathsf{edge\text{-}loss} &= \sum_{i=1,2,\ldots} \mathsf{edge\text{-}loss}_i && (5.3.12)\\
&= \sum_{i=1,2,\ldots} \sum_{t=1,\ldots,x_i} \mathsf{cap}(S_i^t, V_i^t) && (5.3.13)\\
&\le \sum_{i=1,2,\ldots} x_i \cdot \hat{c} && (5.3.14)\\
&\le \frac{2\mathsf{LOSS} \cdot \hat{c}}{C^0 - 2\hat{c}}. && (5.3.15)
\end{aligned}$$

□

5.3.2 Bound the Flow Lost Due to Min-Cut Processing

Lemma 5.2. *The total flow of $\bar{f}$ that we lose from min-cut processing is*

$$\textsf{flow-loss}_1 \le \frac{2\textsf{LOSS} \cdot \hat{c}}{C^0 - 2\hat{c}}(2\lambda(n) + 1/2). \tag{5.3.16}$$

Proof. For a set of nodes $S_i^t \in V_i^0, \forall t = 1, \ldots, x_i$, in $G_i^0 = (V_i^0, E_i^0)$, we denote the size of cut $(S_i^t, V_i^0 \setminus S_i^t)$ with $\mathcal{B}_i^t = \mathsf{cap}(S_i^t, V_i^0 \setminus S_i^t)$. $\mathcal{B}_i^t$ determines the amount of flow of $\bar{f}$ that we take away from $\gamma(G_i)$ as we remove S_i^t from G_i as the smaller side of a min-cut (S_i^t, V_i^t) in G^{t-1}.

A closer examination of the above cutting process shows that

$$\sum_{t=1,2,\ldots,x_i} \mathcal{B}_i^t \leq 2\textsf{edge-loss}_i, \tag{5.3.17}$$

and

$$\sum_{i=1,2,\ldots} \sum_{t=1,2,\ldots,x_i} \mathcal{B}_i^t \leq 2\textsf{edge-loss}, \tag{5.3.18}$$

since the edges in $\mathcal{B}_i^t$ come either from previous min-cuts: $\{(S_i^j, V_i^j), \forall j < t\}$, or from new edges that contribute to $\{(S_i^t, V_i^t)\}$; in addition, each edge e counted in $\textsf{edge-loss}_i$ can be used at most twice toward $\sum_{t=1}^{x_i} \mathcal{B}_i^t$, once for each of the two neighboring sets in $\{S_i^t, t = 1, \ldots, x_i\}$ that share $e \in G_i^0$.

Hence fix $\mathcal{B}_i^t$ for some t. We now calculate the amount of flows of $\bar{f}$ that we lose by cutting off S_i^t. The flow that we lose falls into one of the four types:

(1) flow whose paths are entirely contained in the subgraph of G_i induced by S_i^t;

(2) flow that has to go through edges that are counted in $\mathcal{B}_i^t$, but not counted in (S_i^t, V_i^t);

(3) flow that has to cross (S_i^t, V_i^t) with at least one endpoint in S_i^t;

(4) flow with both endpoints $u'v' \in V_i^t$ such that the flow path intersects the min-cut (S_i^t, V_i^t) at least twice.

Flow of type 1 is counted in $\sum_{u \in S_i^t} \gamma(u, G_i)$ twice. Flow of type 2 has been counted before when S_i^j were cut off for some $j < t$. Flow of type 3 contributes its flow amount once to $\sum_{u \in S_i^t} \gamma(u, G_i)$ and once to the usage of $\textsf{cap}(S_i^t, V_i^t)$. Flow of type 4 are counted twice in the usage of $\textsf{cap}(S_i^t, V_i^t)$.

Note that flow that crosses cut (S_i^t, V_i^t) either has been counted in $\sum_{u \in S_i^t} \gamma(u, G_i)$ at least once or it crosses (S_i^t, V_i^t) at least twice. Hence $1/2(\sum_{u \in S_i^t} \gamma(u, G_i) + \textsf{cap}(S_i^t, V_i^t))$ upper bounds the amount of flow that we lose from $\bar{f}$, that has not been counted earlier, due to cutting off induced subgraph of S_i^t from G_i^{t-1}:

$$\begin{aligned}
\frac{1}{2}\sum_{u\in S_i^t}\gamma(u,G_i)+\frac{1}{2}\mathsf{cap}(S_i^t,V_i^t) &\leq \frac{1}{2}\vec{\pi}_i(S_i^t\cap X_i^0)\lambda(n)+\frac{1}{2}\mathsf{cap}(S_i^t,V_i^t)\\
&\leq \mathsf{cap}(S_i^t,V_i^0\setminus S_i^t)\lambda(n)+\frac{1}{2}\mathsf{cap}(S_i^t,V_i^t)\\
&\leq \mathcal{B}_i^t\lambda(n)+\frac{1}{2}\mathsf{cap}(S_i^t,V_i^t),
\end{aligned}$$

where second inequality is due to the fact that X_i^0 is $\vec{\pi}_i$-flow-linked and Proposition 4.1, which implies that X_i^0 is $\vec{\pi}_i/2$-cut-linked in G_i^0.

Sum over all $S_i^t, \forall t$, we obtain the total flow lost:

$$\begin{aligned}
\mathsf{flow\text{-}loss}_1 &= \sum_{i=1,2,\ldots}\sum_{t=1,\ldots,x_i}(\mathcal{B}_i^t\lambda(n)+\frac{1}{2}\mathsf{cap}(S_i^t,V_i^t)) &(5.3.19)\\
&\leq 2\mathsf{edge\text{-}loss}\cdot\lambda(n)+\frac{1}{2}\mathsf{edge\text{-}loss} &(5.3.20)\\
&\leq \frac{2\mathsf{LOSS}\cdot\hat{c}}{C^0-2\hat{c}}(2\lambda(n)+1/2). &(5.3.21)
\end{aligned}$$

□

Let $1/a_0$ denote the ratio of amount of flow of $\bar{f}$ that we lose during min-cut processing with respect to LOSS in CKS flow decomposition:

$$\frac{\mathsf{flow\text{-}loss}_1}{\mathsf{LOSS}}\leq\frac{1}{a_0}. \tag{5.3.22}$$

Thus we require

$$\frac{\mathsf{flow\text{-}loss}_1}{\mathsf{LOSS}}\leq\frac{(2\lambda(n)+1/2)\cdot 2\hat{c}}{C^0-2\hat{c}}=\frac{(4\lambda(n)+1)\cdot\hat{c}}{C^0-2\hat{c}}\leq\frac{1}{a_0}. \tag{5.3.23}$$

Given an a_0, in order to satisfy (5.3.23), we require

$$C^0\geq(4a_0\lambda(n)+a_0+2)\cdot\hat{c}. \tag{5.3.24}$$

Plugging (5.3.24) in (5.3.12), we obtain the following bound on edge loss due to post-processing of G_i:

$$\textsf{edge-loss} \leq \frac{2\textsf{LOSS} \cdot \hat{c}}{C^0 - 2\hat{c}} \leq \frac{2\textsf{LOSS} \cdot \hat{c}}{a_0(4\lambda(n)+1) \cdot \hat{c}} = \frac{\textsf{LOSS}}{a_0(2\lambda(n)+\frac{1}{2})}. \tag{5.3.25}$$

5.3.3 Obtain the Final Set of Terminals

Recall that $G_i^0 = (V_i^0, E_i^0)$ denote the subgraph G_i we obtain through CKS flow decomposition before any subset of nodes have been removed; $\hat{G}_i = (\hat{V}_i, \hat{E}_i), \forall i$ are the remaining subgraphs of $G_i, \forall i$ at the end of the min-cut processing stage. By (5.3.23), the total flow of $\bar{f}$ that remains is the sum of flow of $\bar{f}$ induced in $\hat{G}_i$, across all i,

$$\begin{aligned} \textsf{remaining-flow} &= \sum_{i=1,2,\ldots} \gamma(\hat{G}_i) && (5.3.26) \\ &\geq \frac{\textsf{OPT}^*(\mathcal{G},\mathcal{T})}{2} - \textsf{flow-loss}_1 && (5.3.27) \\ &\geq \frac{1}{2}\textsf{OPT}^*(\mathcal{G},\mathcal{T})(1-\frac{1}{a_0}), && (5.3.28) \end{aligned}$$

where $\textsf{flow-loss}_1 = \textsf{LOSS}/a_0$ and $\textsf{LOSS} \leq \textsf{OPT}^*(\mathcal{G},\mathcal{T})/2$.

In the sparsest-cut processing, we remove $P_i^1, P_i^2, \ldots, P_i^{y_i}$ from the graph $\hat{G}_i$ that do not meet a certain sparsest cut condition. In the end, we have a subgraph G_i' that does meet the sparsest cut condition on the demands in the remaining subgraph. Now we assign a zero weight to all vertices in the removed regions to zero out demands on these regions and put $P_i^1, P_i^2, \ldots, P_i^{y_i}$ all back in. This graph $\hat{G}_i$ is only more connected with regard to the non removed demand induced by $\bar{f}$ inside $G_i', \forall i$. Hence we emphasize that $\hat{G}_i, \forall i = 1, \ldots, \ell$ are the set of subgraphs that we pass on to the next stage. We give an algorithm for computing the final disjoint subsets $\mathcal{T}_1', \ldots, \mathcal{T}_\ell'$ of $\mathcal{T}$ such that terminal pairs in $\mathcal{T}_i'$ belong to G_i', and hence $\hat{G}_i, \forall i$, and assigning a positive weight to the set of terminals in $\mathcal{T}_i', \forall i$.

In the rest of this section, we prove Theorem 5.2.

Proof of Theorem 5.2: Given a subgraph $\hat{G}_i = (\hat{V}_i, \hat{E}_i)$, we use the procedure as in Figure 5.3.3 to update $\hat{G}_i$ recursively by muting regions that do not satisfy the sparsest cut condition; by "muting" a region P, we treat nodes in P and their ad-

0. Given a subgraph $\hat{G}_i$.
1. If $\gamma(\hat{G}_i) \leq (a_1/4)\beta\lambda(n)$, $\vec{\pi}'_i(u) = \vec{\pi}'_i(v) = 1$ for some pair $uv \in \mathcal{T}'_i$
   with positive flow in $\hat{G}_i$; and $\vec{\pi}'_i(y) = 0$ for $y \neq u, v$.
   Hence we can just route a unit flow between the chosen pair $uv \in \mathcal{T}'_i$
   along an integral path; such a path exists since $\hat{G}_i$ is a connected component.
2. Suppose that $\gamma(\hat{G}_i) > (a_1/4)\beta\lambda(n)$. For $\mathsf{dem}(u,v) = \gamma(u,\hat{G}_i)\gamma(v,\hat{G}_i)/\gamma(\hat{G}_i)$,
   let f' be the maximum concurrent flow for this instance.
   (a) if $f' \geq f_1$, set $\vec{\pi}'_i(u) = \frac{\gamma(u,\hat{G}_i)}{(a_1/2)\beta\lambda(n)}$ $\forall u \in \hat{V}_i$ and stop.
   (b) else $f' < f_1$, find an approximate sparsest cut such that $\frac{\mathsf{cap}(S,\hat{V}_i \setminus S)}{\mathsf{dem}(S,\hat{V}_i \setminus S)} \leq \beta f'$.
   set $\vec{\pi}'_i(u) = 0$, $\forall u \in S$, and
   shut off edges in $\delta^0(S) = (S, \hat{V}_i \setminus S)$
   so that we recurse on $\hat{G}_i[V(\hat{G}_i) \setminus S]$.
3. End

Figure 5.3.1. **Algorithm** FINDING SPARSEST CUTS

jacent edges as if they were removed from $\hat{G}_i$ during the sparsest-cut processing stage, although in the end, we retain these regions entirely in $\hat{G}_i$. We define the following parameters given a remaining subgraph $\hat{G}^t_i$ of G^i after muting some regions, $P^1_i, \ldots, P^{t-1}_i$.

(1) $\hat{G}^t_i = (\hat{V}^t_i, \hat{E}^t_i)$: the remaining subgraph of $\hat{G}_i$ after muting nodes in $P^1_i, \ldots, P^t_i$ and their adjacent edges. $\hat{V}^t_i = \hat{V}_i \setminus \cup_{j=1,\ldots,t} P^j_i$ is the remaining set of vertices in $\hat{G}_i$ at stage t.

(2) $\delta^t(S) = \mathsf{cap}(S, \hat{V}^t_i \setminus S)$ denotes the size of cut $(S, \hat{V}^t_i \setminus S)$ in subgraph $\hat{G}^t_i$.

(3) $\Delta(S) = \mathsf{cap}(S, V^0_i \setminus S)$ denotes the size of cut $(S, V^0_i \setminus S)$ in subgraph G^0_i.

Given $\hat{G}^t_i$, we try to route the following multicommodity product flow between any unordered pair of vertices u, v:

$$\mathsf{dem}^t(u,v) = \frac{\gamma(u,\hat{G}^t_i)\gamma(v,\hat{G}^t_i)}{\gamma(\hat{G}^t_i)}, \tag{5.3.29}$$

where $\gamma(u,\hat{G}^t_i)$ is the flow of $\bar{f}$ at node $u \in \hat{V}^t_i$ that is induced in $\hat{G}^t_i$.

We define $f_1 = \frac{1}{a_1\beta(G)\lambda(n)}$, where $a_1 > 8$, as the minimum concurrent flow value that one needs to obtain for $\mathsf{dem}^t(u,v)$ in order for subgraph $\hat{G}_i^t$ to satisfy the flow-linked property. When the actual flow value $f' < f_1$, we can find a set P_i^{t+1} such that $\delta^t(P_i^{t+1}) \le \mathsf{dem}^t(P_i^{t+1}, \hat{V}_i^t \setminus P_i^{t+1})\beta f'$. We say P_i^{t+1} does not meet the sparsest cut condition for the demands $\mathsf{dem}^t(u,v)$ in subgraph $\hat{G}_i^t$, and we mute P_i^{t+1} in $\hat{G}_i^t$ and recurse on $\hat{G}_i[\hat{V}_i^t \setminus P_i^{t+1}] = \hat{G}_i^{t+1}$. When the flow value $f' \ge f_1$, we stop the recursion, and assign $\vec{\pi}_i'(u) = \frac{\gamma(u,\hat{G}_i^t)}{(a_1/2)\beta\lambda(n)}$ for all $u \in \hat{V}_i^t$.

Let $G_i' = \hat{G}_i^{y_i} = (\hat{V}_i^{y_i}, \hat{E}_i^{y_i})$ be the remaining subgraph of $\hat{G}_i$ by end of sparsest-cut processing after muting nodes in $P_i^1, \ldots, P_i^{y_i}$ and their adjacent edges. Let the set of terminal pairs $\mathcal{T}_i'$ be the subset of $\mathcal{T}_i$ that are contained in subgraph G_i' and let X_i' be the set of terminals of $\mathcal{T}_i'$.

If $\gamma(G_i') \le (a_1/4)\beta\lambda(n)$ when the algorithm terminates, we have obtained a terminal pair $\mathcal{T}_i'$ to route in G_i' and a weight assignment that satisfy all three conditions in the theorem. And we are done with this subgraph.

When $\gamma(G_i') > (a_1/4)\beta\lambda(n)$, a product flow based on the flow of $\bar{f}$ induced in G_i' is routable with throughput at least $f_1 = \frac{1}{a_1\beta(G)\lambda(n)}$ in G_i', where $a_1 > 8$. Hence by assigning a new weight

$$\vec{\pi}_i'(u) = \frac{\gamma(u,G_i')}{(a_1/2)\beta\lambda(n)}, \tag{5.3.30}$$

for all $u \in V(G_i')$, and $\vec{\pi}_i'(u) = 0$ for all other nodes $u \in \hat{V}_i$ in $\hat{G}_i$, we can define a multicommodity flow problem, where for any unordered pair of vertices $u,v \in V(G_i')$, $\mathsf{dem}^{\vec{\pi}_i'}(u,v) = \vec{\pi}_i'(u)\vec{\pi}_i'(v)/\vec{\pi}_i'(X_i')$, that is feasible in both G' and $\hat{G}_i$. Hence X_i' is $\vec{\pi}_i'$-flow-linked in $\hat{G}_i, \forall i$. Finally, we put $P_i^t, \forall t$ back in $\hat{G}_i$ with zero node weight, while retaining the same weight assignment for nodes in G_i'. Hence the sum of the total weight is:

$$\vec{\pi}_i'(\hat{G}_i) = \vec{\pi}_i'(G_i') = \sum_{u \in G_i'} \frac{\gamma(u,G_i')}{(a_1/2)\beta\lambda(n)} = \frac{\gamma(G_i')}{(a_1/4)\beta\lambda(n)}. \tag{5.3.31}$$

Hence for both terminating conditions of the algorithm, we have

$$\vec{\pi}_i'(G_i') \ge \gamma(G_i')(a_1/4)\beta\lambda(n). \tag{5.3.32}$$

Hence

$$\sum_{i=1}^{\ell} \vec{\pi}'_i(X'_i) \geq \sum_{i=1}^{\ell} \frac{\gamma(G'_i)}{(a_1/4)\beta\lambda(n)} \geq \frac{\mathsf{OPT}^*(\mathcal{G},\mathcal{T})}{16\beta\lambda(n)},$$

where $\sum_{i=1}^{\ell}\gamma(G'_i) \geq \mathsf{OPT}^*(\mathcal{G},\mathcal{T})/4$ by taking $a_0 = 4$ and $a_1 = 16$ in Lemma 5.3 and requiring that $\mathcal{C}^0 > 16\lambda(n)\hat{c}$. □

Hence by the end of sparsest-cut processing, we get a new instance X'_i on $\hat{G}_i = (\hat{V}_i, \hat{E}_i)$ with min-cut at least $\hat{c} = \Omega(\log^3 n)$, such that X'_i is $\vec{\pi}'_i$-flow-linked in $\hat{G}_i$, which can be only more connected than G'_i. We tune two parameters: a_0 and a_1, to balance the the initial node degree requirement and the amount of flow of $\bar{f}$ that we retain by the end of min-cut and sparsest-cut processing stages.

Lemma 5.3. *Given a graph $\mathcal{G}$ with min-cut value $C^0 \geq (4a_0\lambda(n) + a_0 + 2)\hat{c}$, where $a_0 \geq 2$. By the end of sparsest-cut processing, the total amount of flow of $\bar{f}$ that we will pass on to next stage of the algorithm for finding EDP in $\mathcal{G}$ is the sum of flow of $\bar{f}$ induced in G'_i, across all i,*

$$\sum_{i=1,2,\ldots} \gamma(G'_i) \geq \frac{1}{2} OPT^*(\mathcal{G},\mathcal{T}) \left(1 - \frac{1}{a_0} - \frac{1}{2a_0(1-8/a_1)}\right), \tag{5.3.33}$$

where $a_1 > 8$.

Proof. In the beginning of the sparsest-cut processing stage, we have

$$\begin{aligned} \mathsf{remaining\text{-}flow} &= \sum_{i=1,2,\ldots} \gamma(\hat{G}_i) & (5.3.34)\\ &\geq \frac{1}{2}\mathsf{OPT}^*(\mathcal{G},\mathcal{T})(1-\frac{1}{a_0}). & (5.3.35)\end{aligned}$$

Combine this initial condition with Lemma 5.4, we have

$$\mathsf{remaining\text{-}flow} = \sum_{i=1,2,\ldots} \gamma(G'_i) \geq \frac{1}{2}\mathsf{OPT}^*(\mathcal{G},\mathcal{T}) \left(1 - \frac{1}{a_0} - \frac{1}{2a_0(1-8/a_1)}\right), \tag{5.3.36}$$

where $\mathsf{LOSS} \leq \mathsf{OPT}^*(\mathcal{G},\mathcal{T})/2$. □

Lemma 5.4. *The amount of flow that we lose from* $\sum_{i=1,2,\ldots} \gamma(\hat{G}_i)$ *due to sparsest-cut processing is* $\textsf{flow-loss}_2 \leq \frac{\textsf{LOSS}}{2a_0(1-8/a_1)}$, *where* $a_1 > 8$.

Proof. To analyze the amount of flow that we lose from sparsest-cut processing, we use a potential function $\varphi(\hat{G}_i)$ to keep track of the edges of $G_i^0 = (V_i^0, E_i^0)$ that we take away from nodes currently in $\hat{G}_i$, after min-cut and during sparsest-cut processing. Note that those lost edges connect to other nodes in V_i^0 from nodes internal to $\hat{G}_i^t$ at stage t.

The counting process is the following. We start with a component G_i such that some nodes in $\hat{G}_i$ have lost some of their edges right after min-cut processing and

$$\varphi_i^0 = \textsf{edge-loss}_i \geq 0.$$

Let φ_i^t be value of $\varphi(\hat{G}_i)$ after removing t sets of vertices $P_i^1, \ldots, P_i^t$ and their adjacent edges from $\hat{G}_i$. Let P_i^{t+1} be the $(t+1)^{st}$ set of vertices that we shut off from $\hat{G}_i$ because internal boundary capacity of P_i^{t+1} has decreased from $\Delta(P_i^{t+1})$ to $\delta^t(P_i^{t+1}) \leq \textsf{dem}^t(P_i^{t+1}, \hat{V}_i^t \setminus P_i^{t+1})\beta f'$.

We update $\varphi(G_i)$ as the following,

$$\varphi_i^{t+1} = \varphi_i^t - (\Delta(P_i^{t+1}) - \delta^t(P_i^{t+1})) + \delta^t(P_i^{t+1}).$$

Since the credit that a cut puts back is less than the credit that it spent, there is a only finite number y_i of such small cuts. By the end of y_i rounds, there must be non-negative credit in $\varphi(\hat{G}_i)$, since nodes in current $\hat{G}_i$ can never gain any internal edges:

$$\begin{aligned}
\varphi(\hat{G}_i) &= \varphi_i^{y_i} \\
&= \textsf{edge-loss}_i - (\Delta(P_i^1) - \delta^0(P_i^1)) + \delta^0(P_i^1) - (\Delta(P_i^2) - \delta^1(P_i^2)) + \delta^1(P_i^2) \\
&\quad - \ldots - (\Delta(P_i^{y_i}) - \delta^{y_i-1}(P_i^{y_i})) + \delta^{y_i-1}(P_i^{y_i}) \\
&\geq 0.
\end{aligned}$$

Hence by summing above inequalities over all i,

$$\sum_{i=1,2,\ldots} \sum_{j=1,2,\ldots,y_i} (\Delta(P_i^j) - 2\delta^{j-1}(P_i^j)) \leq \sum_{i=1,2,\ldots} \textsf{edge-loss}_i \quad (5.3.37)$$

$$= \textsf{edge-loss} \quad (5.3.38)$$

$$\leq \frac{\textsf{LOSS}}{a_0(2\lambda(n)+\frac{1}{2})}. \quad (5.3.39)$$

.

Fix P_i^{t+1} for some $i,t \in [0,\ldots,y_i-1]$, we have the following two lemmas on $\Delta(P_i^{t+1})$ and $\delta^(P_i^{t+1})$.

Lemma 5.5. *For all i and all* $t \in [0,\ldots,y_i-1]$,

$$\Delta(P_i^{t+1}) = \textsf{cap}(P_i^{t+1}, V_i^0 \setminus P_i^{t+1}) \geq \sum_{u \in P_i^{t+1}} \gamma(u, \hat{G}_i^{t+1})/2\lambda(n).$$

Lemma 5.6. *For all i and all* $t \in [0,\ldots,y_i-1]$, $\delta^t(P_i^{t+1}) \leq \frac{4}{a_1}\Delta(P_i^{t+1})$.

Plugging Lemma 5.6 in 5.3.37, we get

$$\sum_{i=1,2,\ldots} \sum_{t=1,2,\ldots,y_i} \Delta(P_i^j)(1-\frac{8}{a_1}) \leq \sum_{i=1,2,\ldots} \sum_{t=1,2,\ldots,y_i} \Delta(P_i^j) - 2\delta^{t-1}(P_i^t))$$

$$\leq \frac{\textsf{LOSS}}{a_0(2\lambda(n)+\frac{1}{2})}.$$

Hence

$$\sum_{i=1,2,\ldots} \sum_{t=1,2,\ldots,y_i} \Delta(P_i^j) = \frac{\textsf{LOSS}}{a_0(2\lambda(n)+\frac{1}{2})(1-8/a_1)}. \quad (5.3.40)$$

Fix $\hat{G}_i^t = (\hat{V}_i^t, \hat{E}_i^t)$ for some $t \in [1,\ldots,y_i]$. We now calculate the amount of flows of $\bar{f}$ that we lose from $\sum_{i=1,2,\ldots} \gamma(\hat{G}_i)$ by shutting off P_i^{t+1} in $\hat{G}_i$. The flow that we lose falls into one of the four types:

(1) its path are entirely contained in the subgraph of $\hat{G}_i$ induced by nodes in P_i^{t+1};

(2) its path contains edges counted in $\Delta(P_i^{t+1})$ but not those in $\delta^t(P_i^{t+1})$;

(3) its path contains edges counted in $\delta^t(P_i^{t+1})$, but with at least one endpoint in P_i^{t+1};

(4) those flow with both endpoints $u'v' \in \hat{V}_i^{t+1}$, such that its path intersects edges counted in $\delta^t(P_i^{t+1})$ for at least twice.

Flow of type 1 contributes to the sum $\sum_{u\in P_i^{t+1}} \gamma(u,\hat{G}_i^t)$ twice. Flow of type 3 contribute its flow value to $\sum_{u\in P_i^{t+1}} \gamma(u,\hat{G}_i^t)$ once and to the usage of $\delta^t(P_i^{t+1}) = \mathsf{cap}(P_i^{t+1},\hat{V}_i^{t+1})$ at least once. Flow of type 4 contribute its flow amount at least twice to the usage of $\mathsf{cap}(P_i^{t+1},\hat{V}_i^{t+1})$. Flow of type 2 has been counted before when P_i^j were mute for some $j \le t$ from $\hat{G}_i$. Note that those flow that crosses (P_i^{t+1},V_i^{t+1}) either has been counted in $\sum_{u\in P_i^{t+1}} \gamma(u,G_i)$ at least once or it goes through the cut (P_i^{t+1},V_i^{t+1}) in $\hat{G}_i^t$ at least twice.

Hence the total amount of flow of $\bar{f}$ that we lose from $\gamma(\hat{G}_i)$, that has not been counted in earlier stages than t, by muting the induced subgraph of P_i^{t+1} and its adjacent edges in $\hat{G}_i^t$:

$$\begin{aligned}
\frac{1}{2}\Big(\sum_{u\in P_i^{t+1}} \gamma(u,G_i) + \mathsf{cap}(P_i^{t+1},V_i^{t+1})\Big) &= \frac{1}{2}\sum_{u\in P_i^{t+1}} \gamma(u,G_i) + \frac{1}{2}\delta^t(P_i^{t+1}) \\
&\le \Delta(P_i^{t+1})\lambda(n) + \frac{1}{2}\delta^t(P_i^{t+1}) \\
&\le \Delta(P_i^{t+1})(\lambda(n) + 2/a_1),
\end{aligned}$$

where the last two inequalities are due to Lemma 5.5 and 5.6.

Summing over all $P_i^t, \forall t, \forall i$, given that $a_1 \ge 8$, the total flow lost in sparsest-cut processing stage is

$$\begin{aligned}
\mathsf{flow\text{-}loss}_2 &= \sum_{i=1,2,\ldots} \sum_{t=1,\ldots,y_i} \Delta(P_i^t)\lambda(n) + \frac{1}{2}\delta^{t-1}(P_i^t) \\
&\le \sum_{i=1,2,\ldots} \sum_{t=1,\ldots,y_i} \Delta(P_i^t)(\lambda(n) + 2/a_1) \\
&\le \frac{(\lambda(n)+2/a_1)\mathsf{LOSS}}{2a_0(\lambda(n)+1/4)(1-8/a_1)} \\
&\le \frac{\mathsf{LOSS}}{2a_0(1-8/a_1)}.
\end{aligned}$$

□

Proof of Lemma 5.5: Given that $\sum_{u\in P_i^{t+1}} \gamma(u,\hat{G}_i^t) \leq \sum_{v\in \hat{V}_i^{t+1}} \gamma(v,\hat{G}_i^t)$ and $\sum_{u\in P_i^{t+1}} \gamma(u,\hat{G}_i^t) + \sum_{v\in \hat{V}_i^{t+1}} \gamma(v,\hat{G}_i^t) = \sum_{u\in \hat{V}_i^t} \gamma(u,\hat{G}_i^t)$, we have

$$\sum_{u\in P_i^{t+1}} \gamma(u,\hat{G}_i^t) \leq \frac{1}{2}\sum_{u\in \hat{V}_i^t} \gamma(u,\hat{G}_i^t). \tag{5.3.41}$$

Next let us define a_2 as the additional flow of $\bar{f}$ for node u in G_i^0 as compared to that in subgraph $\hat{G}_i^t$,

$$\sum_{u\in P_i^{t+1}} \gamma(u,G_i^0) = \sum_{u\in P_i^{t+1}} \gamma(u,\hat{G}_i^t) + a_2. \tag{5.3.42}$$

Since each unit of flow of a_2 uses at least one unit capacity from edges that connect two set of vertices P_i^{t+1} and $V_i^0 \setminus \hat{V}_i^t$ in G_i^0, we have

$$a_2 \leq \Delta(P_i^{t+1}) - \delta^t(P_i^{t+1}). \tag{5.3.43}$$

In addition, we know that

$$\sum_{u\in X_i^0} \gamma(u,G_i^0) \geq \sum_{u\in \hat{V}_i^t} \gamma(u,G_i^0) \geq \sum_{u\in \hat{V}_i^t} \gamma(u,\hat{G}_i^t) + a_2. \tag{5.3.44}$$

Thus we have

$$\begin{aligned}
\sum_{u\in P_i^{t+1}} \gamma(u,G_i^0) &= \sum_{u\in P_i^{t+1}} \gamma(u,\hat{G}_i^t) + a_2 && (5.3.45)\\
&\leq \frac{\sum_{u\in \hat{V}_i^t} \gamma(u,\hat{G}_i^t) + a_2}{2} + \frac{a_2}{2} && (5.3.46)\\
&\leq \sum_{u\in \hat{V}_i^t} \frac{1}{2}\gamma(u,G_i^0)/2 + \frac{a_2}{2} && (5.3.47)\\
&\leq \sum_{u\in X_i^0} \frac{1}{2}\gamma(u,G_i^0)/2 + \frac{a_2}{2}. && (5.3.48)
\end{aligned}$$

Now if $\sum_{u\in P_i^{t+1}} \gamma(u,G_i^0) \le \frac{1}{2}\sum_{u\in V_i^0}\gamma(u,G_i^0)$, i.e. $\vec{\pi}_i(P_i^{t+1}\cap X_i^0) \le \frac{1}{2}\vec{\pi}_i(X_i^0)$,

$$\begin{aligned}
\Delta(P_i^{t+1}) &\ge \mathsf{cap}(P_i^{t+1}, V_i^0 \setminus P_i^{t+1}) \\
&\ge \frac{1}{2}\vec{\pi}_i(P_i^{t+1}\cap X_i^0) \\
&\ge \frac{\sum_{u\in P_i^{t+1}}\gamma(u,G_i^0)}{2\lambda(n)} \\
&\ge \frac{\sum_{u\in P_i^{t+1}}\gamma(u,\hat{G}_i^t)}{2\lambda(n)}.
\end{aligned}$$

Otherwise, $\sum_{u\in P_i^{t+1}}\gamma(u,G_i^0) \ge \frac{1}{2}\sum_{u\in V_i^0}\gamma(u,G_i^0)$. First we have

$$\sum_{u\in V_i^0\setminus P_i^{t+1}} \gamma(u,G_i^0) \ge \sum_{u\in V_i^0}\gamma(u,G_i^0)/2 - a_2/2 \tag{5.3.49}$$

because of (5.3.47) and

$$\sum_{u\in P_i^{t+1}}\gamma(u,G_i^0) + \sum_{u\in V_i^0\setminus P_i^{t+1}}\gamma(u,G_i^0) = \sum_{u\in V_i^0}\gamma(u,G_i^0). \tag{5.3.50}$$

Therefore, we have

$$\begin{aligned}
\Delta(P_i^{t+1}) &= \mathsf{cap}(P_i^{t+1}, V_i^0\setminus P_i^{t+1}) \\
&\ge \frac{1}{2}\vec{\pi}_i((V_i^0\setminus P_i^{t+1})\cap X_i^0) \\
&= \frac{\sum_{u\in V_i^0\setminus P_i^{t+1}}\gamma(u,G_i^0)}{2\lambda(n)} \\
&\ge \frac{(\sum_{u\in V_i^0}\gamma(u,G_i^0)/2 - a_2/2}{2\lambda(n)} \\
&\ge \frac{\sum_{u\in P_i^{t+1}}\gamma(u,G_i^0) - a_2}{2\lambda(n)} \\
&= \frac{\sum_{u\in P_i^{t+1}}\gamma(u,\hat{G}_i^t)}{2\lambda(n)},
\end{aligned}$$

where the last three (in)equalities are due to (5.3.49), (5.3.47), and (5.3.42), and in this order. □

Next, let us bound the size of $\delta^t(P_i^{t+1})$.

Proof of Lemma 5.6: Given that $\sum_{u\in P_i^{t+1}} \gamma(u,\hat{G}_i^t) \le \sum_{v\in \hat{V}_i^{t+1}} \gamma(v,\hat{G}_i^t)$ and $2\gamma(\hat{G}_i^t) = \sum_{u\in P_i^{t+1}} \gamma(u,\hat{G}_i^t) + \sum_{v\in \hat{V}_i^{t+1}} \gamma(v,\hat{G}_i^t)$, we have $\sum_{v\in \hat{V}_i^{t+1}} \gamma(v,\hat{G}_i^t) \le 2\gamma(\hat{G}_i^t)$.

By terminating condition 2(b) in Figure 5.3.3, we have

$$\begin{aligned}
\delta^t(P_i^{t+1}) &\le \mathsf{dem}^t(P_i^{t+1}, \hat{V}_i^t \setminus P_i^{t+1})\beta f_1 \\
&\le \mathsf{dem}^t(P_i^{t+1}, \hat{V}_i^t \setminus P_i^{t+1}) \frac{1}{a_1\lambda(n)} \\
&= \frac{\sum_{u\in P_i^{t+1}} \gamma(u,\hat{G}_i^t)\cdot \sum_{v\in \hat{V}_i^{t+1}} \gamma(v,\hat{G}_i^t)}{a_1\lambda(n)\cdot\gamma(\hat{G}_i^t)} \\
&\le \frac{2\sum_{u\in P_i^{t+1}} \gamma(u,\hat{G}_i^t)}{a_1\lambda(n)} \\
&\le \frac{4}{a_1}\left(\frac{\sum_{u\in P_i^{t+1}} \gamma(u,\hat{G}_i^t)}{2\lambda(n)}\right) \le \frac{4}{a_1}\Delta(P_i^{t+1}),
\end{aligned}$$

where $\mathsf{dem}^t(P_i^{t+1}, \hat{V}_i^t \setminus P_i^{t+1})$ is defined in (5.3.29). □

6 Finding Disjoint Paths in Decomposed Graphs

6.1 Outline of Routing in a Decomposed Subgraph G

We assume that we have the $\vec{\pi}$-cut-linked subgraphs given by Theorem 4.2. We will treat each subgraph and its induced subproblem (G,T) independently. We use $\vec{\pi}(G)$ to denote $\vec{\pi}(V(G))$ in the following sections. Let X be the set of terminals of T that is assigned with a positive weight by function $\vec{\pi}$ in instance G. We further assume that $\vec{\pi}(G) = \Omega(\log^7 n)$. If not, we just route an arbitrary pair of terminals in T; otherwise, we use PROCEDURE EMBEDANDROUTE$(G,T,\vec{\pi})$ in Figure 6.1.1 to route. We first specify a few more parameters and conditions related to (G,T); We then state Theorem 6.1, which we prove through the rest of the paper. Combining Theorem 6.1 and Theorem 4.2 proves Theorem 6.2.

6.1.1 Parameters and Conditions Regarding Subproblem (G,T)

- sampling probability $p = 12(\ln n)/\varepsilon^2\kappa = 1/(\omega \log^2 n + 1)$
- number of split subgraphs $Z = 1/p = \omega \log^2 n + 1$
- $W = (\omega \log^2 n + 1)/(1-\varepsilon)$, for some $\varepsilon < 1$;
- $r \geq \max\{1, (\vec{\pi}(G) - (W-1))/(2W-1)\}$, such that $\forall i \in [1,\ldots,r], 2W-1 \geq \vec{\pi}(X_i) = \sum_{v\in X_i} \vec{\pi}(v) \geq W$ and $\vec{\pi}(\mathcal{X}) \geq \vec{\pi}(G) - (W-1)$: i.e., at most $W-1$ unit of weight is not counted in $\mathcal{X}$.

Given graph G with min-cut $\Omega(\log^3 n)$ and a weight function $\vec{\pi} : V(G) \to \mathbb{R}^+$
$\{G^1, \ldots, G^Z\}$ = $\textsc{Split}(G, Z, \vec{\pi})$
$\{\mathcal{X}, \mathcal{C}\}$ = $\textsc{Clustering}(G^Z, \vec{\pi})$, where $\mathcal{X} = \{X_1, \ldots, X_r\}$ and $\mathcal{C} = \{C_1, \ldots, C_r\}$
Given a set of superterminals $\mathcal{X}$ of size r
Let $\mathcal{X}$ map to vertex set $V(H)$ of Expander H
For $t = 1$ to $\omega \log^2 n$
    $(S, \bar{S} = \mathcal{X} \setminus S)$ = KRV-$\textsc{FindCut}(\mathcal{X}, \{M_k : k < t\})$ s. t. $|S| = |\bar{S}| = r/2$
    Matching M_t = $\textsc{FindMatch}(S, \bar{S}, G^t)$ s.t. M_t is routable in G^t
Combine $M_1, \ldots, M_{\omega \log^2 n}$ to form the edge set F on vertices $V(H)$
$\textsc{ExpanderRoute}(H, T, X)$
End

Figure 6.1.1. **Procedure** $\textsc{EmbedAndRoute}(G, T, \vec{\pi})$

6.1.2 Two Theorems to Prove

Theorem 6.1. *Given an induced instance (G, T) with min-cut of G being $\Omega(\log^3 n)$ and a weight function $\vec{\pi} : V(G) \to \mathbb{R}^+$ such that X is $\vec{\pi}$-cut-linked in G and $\vec{\pi}(G) = \Omega(\log^7 n)$,* $\textsc{EmbedAndRoute}$ *routes at least* $\max\{1, \Omega(\vec{\pi}(G)/\log^7 n)\}$ *pairs of T in G edge disjointly.*

Theorem 6.2. *Given an EDP instance $(\mathcal{G}, \mathcal{T})$, where $\mathcal{G}$ has a min-cut $\Omega(\lambda(n)\kappa)$, we can route $\Omega(\mathit{OPT}^*(\mathcal{G}, \mathcal{T})/f)$ terminal pairs edge disjointly in $\mathcal{G}$, where the approximation factor f is $O(\lambda(n)\beta(\mathcal{G})W \log^5 n)$.*

6.2 Obtaining Z Split Subgraphs of G

In this section, we analyze a procedure that splits a graph G, with min-cut $\kappa = \Omega(\log^3 n)$, into Z subgraphs b extending a uniform sampling scheme from Karger [1994]. We thus obtain a set of cut-linked instances as in Lemma 6.1, which immediately follows from Theorem 6.3.

Procedure Split$(G, Z, \vec{\pi})$: Given a graph $G = (V, E)$ with min-cut $\kappa = \Omega(\log^3 n)$, a weight function $\vec{\pi} : V(G) \to \mathbb{R}^+$, a set of terminals X in G such that (G, X) is a $\vec{\pi}$-cut-linked instance, and probability $p = 1/Z$.

Output: A set of randomized split subgraphs $G^1,\ldots,G^Z$ of G.
Each split subgraph $G^j, \forall j = 1,\ldots,Z$ inherits the same set of vertices of G; Edges of G are placed independently and uniformly at random into the Z subgraphs; each $e = (u,v) \in E$ is placed between the same endpoints u,v in the chosen subgraph. We retain the same weight function $\vec{\pi}$ for all nodes in V in each split subgraph $G^j, \forall j$.

Lemma 6.1. *With high probability, X is $\frac{(1-\varepsilon)\vec{\pi}}{Z}$-cut-linked in $G^j, \forall j$, for some $\varepsilon < 1$.*

Proof. Since X is $\vec{\pi}$-cut-linked in G hence $|\delta(S)| \geq \vec{\pi}(S \cap X)$, $\forall S$ such that $\vec{\pi}(S \cap X) \leq \vec{\pi}(X)/2$ in G. Let $\delta_j(S)$ denote the size of cut $(S, V \setminus S)$ in G^j. With probability $1 - O(\log^2 n/n^2)$, we have $\left|\delta_j(S)\right| \geq (1-\varepsilon)p\,|\delta(S)| \geq (1-\varepsilon)p\vec{\pi}(S \cap X))$, for all S such that $\vec{\pi}(S \cap X) \leq \vec{\pi}(X)/2$ and all j as shown in Theorem 6.3. Hence X is $(1-\varepsilon)\vec{\pi}/Z$-cut-linked in $G^j, \forall j$. □

Theorem 6.3 says that all cuts can be preserved in all split graphs $G^1,\ldots,G^Z$ of G we thus obtain. Recall for $S \in V$, $|\delta_G(S)|$ denote the size of $(S, V \setminus S)$ in G. For the same cut $(S, V \setminus S)$, we have $\mathbf{E}[|\delta_{G^j}(S)|] = p\,|\delta_G(S)|$ in $G^j, \forall j$, where p is the probability that an edge $e \in E$ is placed in $G^j, \forall j$.

Theorem 6.3. *Let $G = (V,E)$ be any graph with unit-weight edges and min cut κ. Let $\varepsilon = \sqrt{3(d+2)(\ln n)/p\kappa}$. If $\varepsilon \leq 1$, then with probability $1 - O(\log^2 n/n^d)$, every cut $(S, V \setminus S)$ in every subgraph $G^1, G^2, \ldots, G^Z$ of G has value between $(1-\varepsilon)$ and $(1+\varepsilon)$ times its expected value $p\,|\delta_G(S)|$.*

We give an overview of our proof by introducing a definition by Karger [1994], regarding a uniform random sampling scheme on an unweighted graph $G = (V,E)$; Lemma 6.2 immediately follows from this definition. We then state Karger's theorem regarding preserving all cuts of G in a sampled subgraph, under a certain min-cut condition. Finally, we show the details of our proof to Theorem 6.3. For the sake of completeness, we also give Karger's proof to Theorem 6.4.

Definition 6.1. **(Karger [1999])** *A p-skeleton of G is a random subgraph $G(p)$ constructed on the same vertices of G by placing each edge $e \in E$ in $G(p)$ independently with probability p.*

Lemma 6.2. *Every randomized subgraph $G^j, \forall j$, is a p-skeleton of G.*

Proof. Recall the construction of a random subgraph $G^j, \forall j$, of G: on the same set of vertices as G, each edge $e \in E$ of the original graph G is placed in G^j independently with probability p. Hence, $G^j, \forall j$, is a p-skeleton of G by Definition6.1. $\square$

Theorem 6.4. **(Karger [1999])** *Let G be a graph with unit-weight edges and min-cut κ. Let $p = 3(d+2)(\ln n)/\varepsilon^2\kappa$. With probability $1 - O(1/n^d)$, every cut in a p-skeleton of G has value between $(1-\varepsilon)$ and $(1+\varepsilon)$ times its expected value.*

Proof of Theorem 6.3: Define an indicator variable $X_e^j, \forall j, \forall e \in E$, such that $X_e^j = 1$ when e is placed in G^j, and 0 otherwise; hence X_e^j is a Bernoulli random variable with success probability p, $\forall j, \forall e$. Note that random variables $X_e^j, \forall j = 1, \ldots, 1/p$, are not independent; in fact, $\sum_{j=1}^{1/p} X_e^j = 1$ for all e.

Consider a cut $(S, \bar{S})$ of size c in G. Let $X_1^j, X_2^j, \ldots, X_c^j$ be the indicator variables that signal whether edges $e_1, e_2, \ldots, e_c$ of cut $(S, \bar{S})$ appear in G^j. Define $X_j(S, \bar{S}) = \sum_{y=1}^{c} X_y^j$ as the size of the cut in a random subgraph G^j of G. Given that $X_1^j, X_2^j, \ldots, X_c^j$ are i.i.d. random variables whose common distribution is the Bernoulli distribution with parameter p, we can apply Chernoff bound to obtain the following lemma.

Lemma 6.3. *Consider a cut $(S, V \setminus S)$ of size c in unweighted graph $G = (V, E)$. Let $X_j(S, \bar{S})$ be the size of the corresponding cut in a randomized split graph G^j. Then $\forall S$, $\forall j$, we have*

$$\mathbf{Pr}\left[\left|X_j(S, V \setminus S) - pc\right| \geq \varepsilon pc\right] \leq 2e^{-\varepsilon^2 pc/3}. \tag{6.2.1}$$

Lemma 6.4. **Chernoff [1952])** *Let X be a sum of independent Bernoulli random variables with success probability $p_1, \ldots, p_m$ and expected value $\mu = \sum p_i$. Then for $\varepsilon \leq 1$*

$$\mathbf{Pr}[|X - \mu| \geq \varepsilon\mu] \leq 2e^{-\varepsilon^2\mu/3}.$$

Let $r = 2^n - 2$ be the number of cuts in graph G, and hence $G^1, \ldots, G^Z$, and let $c_1, \ldots, c_r$ be the expected values of the r cuts in a p-skeleton listed in nondecreasing order so that $p\kappa = c_1 \leq c_2, \ldots, \leq c_r$. Given a split graph $G^j, \forall j$, let $\mathcal{E}_k^j, \forall j, \forall k$ be the event that the value of a cut $X_j(S, V/S)$ in G^j diverges from its expectation c_k

by more than εc_k. First we have $\mathbf{Pr}\left[\mathcal{E}_k^j\right] \leq 2e^{-\varepsilon^2 c_k/3}$ by Lemma 6.3. We then apply a union bound to sum up (6.2.1) for all r cuts in $G^j, \forall j$.

Given that every random split subgraph $G^j, \forall j$, is a p-skeleton of G by Lemma 6.2, we apply Karger's statement as in the form below, to all subgraphs G^j with the following parameters: $p = 12(\ln n)/\varepsilon^2\kappa$ and $\kappa = 12(\ln n)(\omega \log^2 n + 1)/\varepsilon^2$ for a given ε; following Karger's proof to Theorem 6.4, we have:

Lemma 6.5. **(Karger [1999])** $\forall G^j$, $\sum_{k=1}^{r} \mathbf{Pr}\left[\mathcal{E}_k^j\right] \leq O(1/n^d)$.

We can then use a union bound to sum up probabilities of bad events across all split subgraphs $G^1, \ldots, G^Z$ of G, which yields following:

$$\sum_{j=1}^{Z} \sum_{k=1}^{r} \mathbf{Pr}\left[\mathcal{E}_k^j\right] \leq O(\log^2 n/n^d) \tag{6.2.2}$$

Note that $\mathcal{E}_k^j, \forall j = 1, \ldots, Z$, are not independent, since the indicator random variables that contribute to value of $X_j(S, V/S)$ are not at all independent across all subgraphs. However, we only use a union bound that does not assume anything about dependency among events.□

Proof of Theorem 6.4: **(Karger [1999])** We give a sketch of Karger's proof as shown in Karger [1999] here for the sake of completeness. To prove Theorem 6.4, Karger uses a union bound to show that the sum of probabilities of all *bad* events in a p-skeleton of G is $O(1/n^d)$, where a bad event refers to some cut in a p-skeleton of G diverges from its expected value k by more than εk. The proof of this claim follows by using two lemmas:

Lemma 6.6. **(Karger [1999])** *In an undirected graph, the number of* α*-minimum cuts is less than* $n^{2\alpha}$.

The "expected value" graph $\bar{G}$ of $G^j, \forall j$, is a weighted graph with all vertices and edges of the original unweighted graph $G = (V, E)$, and with edge weight p assigned to edge $e, \forall e \in E$. Note that the minimum cut of $\bar{G}$ is $p\kappa$, where κ is the minimum cut of G. Lemma 6.6 applied to $\bar{G}$, the "expected value" graph of a p-skeleton of G, states that the number of cuts within α factor of the minimum $p\kappa$ increases exponentially with α. On the other hand, the Chernoff bound says that one such cut diverges too far from its expected value decreases exponentially with α as shown in (6.2.1). Combining these two lemmas and balancing the exponential rates proves Theorem 6.4 using a union bound. □

6.3 Forming Superterminals that are Well-Linked

The procedure in this section constructs superterminals as follows. It finds connected subgraphs C in G^Z, where $\vec{\pi}(C) = \Omega(\log^2 n)$, each connecting a subset of terminals. Roughly, the idea is that these clustered terminals are better connected than individual terminals. They are well linked in the sense that any cut that splits off K superterminals as one entity contains at least K edges in $G^j, \forall j$ This allows us to compute congestion-free maximum flows in Section 6.4.1.

Given split subgraphs $G^1, \ldots, G^Z$ of G, each with the same weight function $\vec{\pi}$ on its vertex set $V(G^j) = V, \forall j$, that we obtain through PROCEDURE SPLIT$(G, Z, \vec{\pi})$, we aim to find a set $\mathcal{X} = \{X_1, \ldots, X_r\}$ of node-disjoint "superterminals", where each superterminal $X_i \in \mathcal{X}$ consists of a subset of terminals in X and each X_i gathers a weight between W and $2W-1$. In addition, we want to find an edge-disjoint set of clusters $\mathcal{C} = \{C_1, \ldots, C_r\}$, where $C_i = (V_i, E_i)$, such that $X_i \subseteq V_i$ and C_i is a connected component, and hence all nodes in X_i are connected through E_i. W.l.o.g., we pick G^Z for forming such clusters $C_i, \forall i$; note that G^Z is a connected graph with a min-cut of $\Omega(\log n), whp$, by Theorem 6.3.

Procedure Clustering$(G^Z, \vec{\pi})$**:** Given a split subgraph G^Z and a weight function $\vec{\pi} : V(G^Z) \to \mathbb{R}^+$ and $\vec{\pi}(V(G^Z)) = \vec{\pi}(G) \geq W$.
Output: $\mathcal{X} = \{X_1, \ldots, X_r\}$ and $\mathcal{C} = \{C_1, \ldots, \mathcal{C}_r\}$ as specified in Lemma 6.7.
We group subsets of vertices of V in an edge-disjoint manner, following a procedure from Chekuri et al. [2004a], by choosing an arbitrary rooted spanning tree of G^Z and greedily partitioning the tree into a set $\mathcal{C}$ of edge-disjoint subgraphs of G^Z.

Lemma 6.7. (CKS2004 Chekuri et al. [2004a]) *Let G^Z be a connected graph with a weight function $\vec{\pi} : V(G^Z) \to [0, W]$ such that $\vec{\pi}(V(G^Z)) \geq W$. We can find $r \geq \max\{1, (\vec{\pi}(G) - (W-1))/(2W-1)\}$ edge-disjoint connected subgraphs, $C_1 = (V_1, E_1), \ldots, C_r = (V_r, E_r)$, such that there exist vertex-disjoint subsets $X_1, \ldots, X_r$ and for each i: (a) $X_i \subseteq V_i$ and (b) $2W - 1 \geq \sum_{v \in X_i} \vec{\pi}(v) \geq W$.*

Result. To get an intuition of the purpose of forming such clusters, consider a cut $(U, V \setminus U)$ in a split subgraph $G^j, \forall j$. Let U be a subset of $V(G)$ such that $\vec{\pi}(U) = \sum_{x \in U \cap X} \vec{\pi}(x) \leq \vec{\pi}(X)/2$. Let K be the number of superterminals that are contained in U. We have the following lemma, which captures the notion of superterminals being "well-linked", with a hint of Definition 4.3.

Lemma 6.8. *$\forall$ split subgraphs $G^1, \ldots, G^Z$, where $Z = \omega \log^2 n + 1$, and $\forall U \subset V(G)$ s.t. $\vec{\pi}(U) \leq \vec{\pi}(X)/2$, $|\delta_{G^j}(U)| \geq K$, where $K = |\{X_i \in \mathcal{X} : X_i \subseteq U\}|$.*

Proof. With high probability, X is $\frac{(1-\varepsilon)\vec{\pi}}{(\omega \log^2 n+1)}$-cut-linked in $G^1, \ldots, G^Z$, as shown in Lemma 6.1. Recall that in our clustering scheme, total weight of all terminals in one cluster is at least $W = \frac{(\omega \log^2 n+1)}{1-\varepsilon}$, then $\forall j$,

$$
\begin{aligned}
|\delta_{G^j}(U)| &\geq \sum_{x \in U} \frac{(1-\varepsilon)\vec{\pi}(x)}{(\omega \log^2 n+1)} \\
&\geq \frac{(1-\varepsilon)}{(\omega \log^2 n+1)} \sum_{i: X_i \subseteq U} \sum_{x \in X_i} \vec{\pi}(x) \\
&\geq \frac{(1-\varepsilon)KW}{(\omega \log^2 n+1)} \\
&\geq K
\end{aligned}
$$

□

6.4 Construct and Embed an Expander H in G

In this section, we use the superterminals from the previous section as nodes in an expander H that we embed in G. The edges of H are defined using a technique in Khandekar et al. [2006] that builds an expander using $O(\log^2 n)$ matchings. We embed this expander in G by routing each matching in one of the split graphs using a maximum flow computation. This allows us to embed H into G with no congestion. The following procedure restates this outline. Theorem 6.5 is a main technical contribution of this paper.

Procedure EmbedExpander$(G^1, \ldots, G^{\omega \log^2 n}, \mathcal{X})$**:**

0. Given a set of points $V(H)$ of size k
1. for $t = 1$ to $\omega \log^2 n$
2. $\quad (S, \bar{S} = V(H) \setminus S)$ = KRV-FINDCUT$(V(H), \{M_k : k < t\})$ s.t. $|S| = |\bar{S}| = k/2$
3. $\quad M_t$ = FINDMATCH$(S, \bar{S})$ s.t. M_t is a matching between S and $\bar{S}$
4. Combine $M_1, \ldots, M_{\omega \log^2 n}$ to form the edge set F on vertices $V(H)$
5. End

Figure 6.4.2. **KRV-Procedure** CONSTRUCTING AN α-EXPANDER H.

Output: An expander $H = (V', F)$ routable in G s.t. $|V'| = r$ and $\forall i \in V'$, $\vec{\pi}(i) = \vec{\pi}(X_i)$ and $\vec{\pi}(H) = \vec{\pi}(\mathcal{X})$; F consists of $M_1, \ldots, M_{\omega \log^2 n}$.
We use Step (3) to (8) of PROCEDURE EMBEDANDROUTE in Figure 6.1.1, where we substitute PROCEDURE FINDMATCH with Figure 6.4.3 while relying on an existing PROCEDURE KRV-FINDCUT Khandekar et al. [2006]. At each round t, we use KRV-FINDCUT to generate an equal-sized partition $(S, \mathcal{X} \setminus S = \bar{S})$; we then find a matching M_t between S and $\bar{S}$ by computing a single-commodity max-flow using FINDMATCH$(S, \bar{S}, G^t)$ in G^t, that we add to F as edges.

Theorem 6.5. *(a)* EMBEDEXPANDER *constructs a* $1/4$*-expander* $H = (V', F)$*; (b) in addition, H is embedded into G as follows. Each node i of H corresponds to a superterminal X_i in $\mathcal{X}$ in G such that all superterminals are mutually node disjoint and each superterminal is connected by a spanning tree, T_i, in G. Each edge (i, j) in H corresponds to a path, P_{ij} from a node in X_i to a node in X_j. All paths P_{ij} and trees T_i are mutually edge disjoint in G.*

Proof. The expander property (a) follows from a result of Khandekar, Rao and Vazirani Khandekar et al. [2006]; they show the procedure in Figure 6.4.2 produces an expander H.

Theorem 6.6. (KRV06 Khandekar et al. [2006]) *Given a set of nodes $V(H)$ of size k, $\exists$ a* KRV-FINDCUT *procedure s.t. given any* FINDMATCH *procedure, the* KRV-PROCEDURE *as in Figure 6.4.2. produces an α-expander graph H, for* $\alpha \geq 1/4$.

Each edge $e = (i, j)$ in the matching M_t maps to an integral flow path that connects X_i and X_j in G^t; all such flow paths can be simultaneously routed in

G^t edge disjointly due to the max-flow computation as we show in Lemma 6.9. Since each matching M^t is on a unique split subgraph G^t, the entire set of edges in $M_1, \ldots, M_{\omega \log^2 n}$, that comprise the edge set F of H, correspond to edge disjoint paths in $G^1, \ldots, G^{Z-1}$, where $Z = \omega \log^2 n + 1$. Finally, all spanning trees $T_i, \forall i$, are constructed using disjoint set of edges in G^Z as in Lemma 6.7. ☐

6.4.1 Finding a Matching through a Max-flow Construction

---

0. Given an equal partition $(S, \bar{S})$ of $\mathcal{X}$, we form a flow graph G' from G^t by adding auxiliary nodes and directed unit-capacity edges:
1. Add a special source and sink nodes s_0 and t_0;
2. Add nodes $s_1, \ldots, s_{r/2}$ and an edge from s_0 to $s_k, \forall k = 1, \ldots, r/2$;
3. Add nodes $t_1, \ldots, t_{r/2}$; from each $t_k, \forall k = 1, \ldots, r/2$, add an edge to t_0
4. From each $s_k, \forall k$, add an edge to each terminal $x \in X_{i_k}$ s.t. $X_{i_k} \in S$
5. To each node t_k, add an edge from each terminal $x \in X_{j_k}$ s.t. $X_{j_k} \in \bar{S}$
6. Route a max-flow from s_0 to t_0
7. Decompose the flow to obtain a matching between S and $\bar{S}$
8. End

Figure 6.4.3. **Procedure** FINDMATCH$(S, \bar{S}, G^t)$

In this section, we show that given an arbitrary equal partition $(S, \bar{S})$ of the set $\mathcal{X} = \{X_1, \ldots, X_r\}$, that we obtain through PROCEDURE CLUSTERING$(G^Z, \vec{\pi})$, we can use the following procedure to route a max-flow of size $r/2$, such that the integral flow paths that we obtain through flow decomposition induce a perfect matching between S and $\bar{S}$. Let $S = \{X_{i_1}, \ldots, X_{i_{r/2}}\}$ and $\bar{S} = \{X_{j_1}, \ldots, X_{j_{r/2}}\}$.

Lemma 6.9. *In each sampled graph G^t,* FINDMATCH *produces a perfect matching M_t between an equal partition $(S, \bar{S})$ of $\mathcal{X}$ such that for each edge in $e = (i, j) \in M_t$, there is an integral unit-flow path P_{ij} from a terminal in $X_i \in S$ to a terminal in $X_j \in \bar{S}$. All paths $P_{ij}, s.t. (i, j) \in M_t$ are edge disjoint in G^t.*

We first prove the following lemma.

Lemma 6.10. *Every $s_0 - t_0$ cut has size at least $r/2$ in the flow graph G'.*

Proof. Let $(U,\bar{U})$ be a cut in the flow graph that separates s_0 from t_0; w.l.o.g., let U be subset such that $\vec{\pi}(U \cap X) \leq \vec{\pi}(X)/2$, and let $s_0 \in U$ (otherwise, we can just rename all the auxiliary nodes and the two subsets S and $\bar{S}$).

Consider any superterminal $X \in \mathcal{X}$ that we obtained through lemma 6.7; if X is contained either in U or $\bar{U}$, we call such a superterminal X uncut; otherwise, we say X is cut by $(U,\bar{U})$.

(1) Let $K_c^s = |\{X \in S : X \cap U, X \cap \bar{U} \neq \emptyset\}|$ denote the number of superterminals in S that is cut by $(U,\bar{U})$.

(2) Let $\overline{K_{uc}^s} = |\{X \in S : X \subseteq \bar{U}\}|$ be the number of superterminals in S that is contained in $\bar{U}$;

(3) Let $K_{uc}^s = |\{X \in S : X \subseteq U\}|$ denote the number of superterminals in S that is contained in U; hence $K_{uc}^s + \overline{K_{uc}^s} + K_c^s = r/2$, where $r = |\mathcal{X}|$.

(4) Let $K_c^t = \left|\{X \in \bar{S} : X \cap U, X \cap \bar{U} \neq \emptyset\}\right|$ denote the number of superterminals in $\bar{S}$ that is cut.

(5) Let $K_{uc}^t = \left|\{X \in \bar{S} : X \subseteq \bar{U}\}\right|$ denote the number of superterminals in $\bar{S}$ that is contained in U.

Given that G is $\vec{\pi}$-cut-linked, we know that the sampled graph G^j is $(1-\varepsilon)\vec{\pi}/(\omega\log^2 n+1)$-cut-linked *whp* by Lemma 6.1. Recall that in our clustering scheme, total weight of all terminals in one superterminal is at least $W = \frac{(\omega\log^2 n+1)}{1-\varepsilon}$. Note that there is at least one directed auxiliary edge crossing the cut for all superterminals except those in S that is contained in U or those in $\bar{S}$ that is contained in $\bar{U}$.

Thus we know

$$
\begin{aligned}
|\delta_{G'}(U)| &\geq |\delta_{G'}(U)| + K_{uc}^t + \overline{K_{uc}^s} + K_c^s + K_c^t \\
&\geq \frac{(1-\varepsilon)\sum_{x\in U}\vec{\pi}(x)}{\omega\log^2 n+1} + K_{uc}^t + \overline{K_{uc}^s} + K_c^s + K_c^t \\
&\geq \frac{(1-\varepsilon)(K_{uc}^s + K_{uc}^t)W}{\omega\log^2 n+1} + K_{uc}^t + \overline{K_{uc}^s} + K_c^s + K_c^t \\
&\geq K_{uc}^s + \overline{K_{uc}^s} + K_c^s \\
&\geq r/2.
\end{aligned}
$$

Hence we have shown that the size of every cut $(U,\bar{U})$ in the flow graph G' has size at least $r/2$. □

Proof of Lemma 6.9: By Lemma 6.10, and the fact that there $\exists$ a $s_0 - t_0$ cut of size $r/2$, (e.g., $(\{s_0\}, V(G') \setminus \{s_0\})$) we know the $s_0 - t_0$ min-cut is $r/2$. Hence by the max-flow min-cut theorem, we know that there $\exists$ a max-flow of size $r/2$ from s_0 to t_0. We next decompose the max-flow into $r/2$ integer flow paths, which induce a perfect matching M_t between S and $\bar{S}$ as follows. Consider an integral flow path $P_k, \forall k = 1,\ldots,r/2$. Let directed path P_k start with s_0 and go through $s_k, x \in X_{i_k} \in S$ for some x; and let P_k end with $y \in X_{j_{k'}} \in \bar{S}, t_{k'}, t_0$ for some $k' \in [1,\ldots,r/2]$ and some terminal y. No other path in the max-flow can go through the same pair of superterminals X_{i_k}, $X_{j_{k'}}$ due to the capacity constraints on edges (s_0, s_k) and $(t_{k'}, t_0)$. Hence $M_t = \{(i_k, j_{k'}), \forall k \in [1,\ldots,r/2],$ where $k' \in [1,\ldots,r/2]\}$ is a perfect matching between S and $\bar{S}$. □

6.5 Routing on an Expander H Node Disjointly

In this section, we show that the following greedy algorithm routes $\Omega(K/\log^5 n)$ pairs of terminals, where $K = |V(H)| = \Omega(\vec{\pi}(G)/W)$, in H.

Procedure ExpanderRoute(H,T,X)**:** Given an uncapacitated expander H with at least $512\log^5 n$ nodes, with node degree $\omega\log^2 n$. While there is a pair (s,t) in $T \subseteq \mathcal{T}$ whose path length is less than D in $H = (V,E)$, where $D = a_3\omega\log^3 n$ and $a_3 = 32$ is a constant; Remove both nodes and edges from H, along a path through which we connect a pair of terminals in T.

Since we take away both nodes and edges as we route a path across the expander H due to the node capacity constraints on $V(H)$, routing the set P of pairs via integral paths on H induces no congestion in G by Theorem 6.5. We now argue that $|P|$ is large to finish our proof. Let H' be the remaining graph of expander $H = (V,E)$, after we take away nodes and edges along the paths used to route P. Note that all remaining pairs $T' \subseteq T$ in H' must have distance at least D. This is the main condition that allows us to prove the following theorem.

Theorem 6.7. *The procedure above routes $\Omega(K/\log^5 n)$ pairs, node disjointly, in degree-$(\omega \log^2 n)$ expander $H = (V,E)$ with $K \geq 512\log^5 n$ nodes.*

We first prove the following lemma regarding a multicut L in H'.

Lemma 6.11. $\exists$ *a multicut of size at most $K/2a_3$ in the remaining graph H' of H.*

Proof. Let us first state the following lemma which follows from arguments of Garg et al. [1996].

Lemma 6.12. *If all remaining terminal pairs in $T' \subseteq T$ have distances at least D in H', then there exists a multicut L in $H' = (V',E')$ of size $|E'|\log n/D$ in H' that separates every source and sink pair $s_i t_i \in T'$.*

Applying Lemma 6.12 to H', we have that there exists a multicut of size at most $K\omega\log^3 n/2D = K/2a_3$ given that $|E'| \leq |E| = K\omega\log^2 n/2$ in the remaining graph H'. $\square$

We prove Theorem 6.7, by noting that condition 1 of Theorem 4.2 implies that any multicut of the terminals in H' ensures that no piece in H' separated by L contains more than half the weight of all terminals in H. We use this fact to show that the multicut L can be rearranged to find a "weight-balanced" cut in H', which corresponds to a node-balanced cut in H. Any node-balanced cut, however, in H must have at least $\Omega(K)$ edges. Using a proper choice of a_3, we force this balanced cut to contain at most half as many edges in H' as in H. Thus, we show $\Omega(K)$ edges have been removed when routing P. Since routing each such pair removes at most $D\omega\log^2 n(O(\log^5 n)$ edges. We conclude $|P|$ must be $\Omega(K/\log^5 n)$.

Proof of Theorem 6.7: Recall that initially $\vec{\pi}(H) = \vec{\pi}(\mathcal{X}) \geq \vec{\pi}(G) - (W-1)$, since at most $W-1$ of $\vec{\pi}(G)$ is not assigned to any node in H, and each node in H has weight between W and $2W-1$ as in shown in proof of Lemma 6.7. Hence the total weight taken away from routing P terminal pairs of distance at most D is at most $DP(2W-1)$.

To facilitate our analysis, we first alter $\vec{\pi}$ slightly to generate a new function $\vec{\pi}'(i), \forall i \in V(H')$,

Procedure Alter($\vec{\pi}$, $\vec{\pi}'$): For a pair of terminals $uv \in T$ such that u takes away a certain amount of weight, we remove the same amount (as specified in condition 1 of Theorem 4.2) from $\vec{\pi}(v)$ if v remains in H' and define this updated weight of H' as $\vec{\pi}'(H')$. Thus we have $\vec{\pi}'(H') \geq \vec{\pi}(G) - (W-1) - 2DP(2W-1)$.

It is easy to see that only remaining pairs $uv \in T'$ contribute a positive weight to $\vec{\pi}'(H')$ according to their flow in $\bar{f}$ like that of condition 1 in Theorem 4.2; hence each connected component in H', separated by multicut L, has a weight of at most $\vec{\pi}'(H')/2$.

Let L be the multicut that separates all remaining terminals pairs $T' \in T$ in H'. L cuts the graph H' and hence group nodes in $V(H')$ into clusters, such that weight of each cluster according to $\vec{\pi}'$ is less than half of the total remaining weight $\vec{\pi}'(H')$ of H', since each pair of terminals that contribute the same amount of weight to $\vec{\pi}'(H')$ must belong to different multicut clusters.

We then use L to find a weight-balanced cut $(U', V' \setminus U')$ in H' such that each side has weight at least $\vec{\pi}'(H')/4$, where $\vec{\pi}'(H') \geq \vec{\pi}(G) - (W-1) - 2(2W-1)D|P|$. It is straightforward to verify that any partition $(U, V(H) \setminus U)$ in H, such that $U' \subseteq U$ and $(V' \setminus U') \subseteq (V(H) \setminus U)$, is node-balanced in H as shown in Lemma 6.13 and Lemma 6.14.

We build a $(1/4, 3/4)$-weight-balanced partition of H' in the following way: start two empty sides A and B, and start adding the connected components (after removing the multicut L) of H' to the smaller side repeatedly. Each component contains at most $\vec{\pi}'(H')/2$ due to condition 1 of Theorem 4.2 and PROCEDURE ALTER; in the end neither side can contain more than $3\vec{\pi}'(H')/4$ of weight; indeed, consider the step where, w.l.o.g, side A were put over $3/4$ of $\vec{\pi}'(H')$ by adding a component d: in that step, d could not have been added to A, since $\vec{\pi}'(A) \geq \vec{\pi}'(H')/4 \geq \vec{\pi}'(B)$ before d were added, given that $d \leq \vec{\pi}'(H')/2$.

Lemma 6.13. *Let $(U', V(H') \setminus U')$ be a $(1/4, 3/4)$-weight-balanced cut in H'. Consider any cut $(U, V(H) \setminus U)$ in H, such that $U' \subseteq U$ and $(V(H') \setminus U') \subseteq (V(H) \setminus U)$ before we route any of the P paths:*

$$\min(|U|, |V \setminus U|) \geq \frac{\vec{\pi}(G) - (W-1)}{4(2W-1)} - DP/2$$

Proof. Indeed, if U is the smaller side, $|U| \geq |U'|$; otherwise, we have $|V(H) \setminus U| \geq |V(H') \setminus U'|$. For both U' and $V(H') \setminus U'$, we have

$$\begin{aligned}
\left|U'\right| &\geq \vec{\pi}'(H')/4(2W-1) \\
\left|V(H') \setminus U'\right| &\geq \vec{\pi}'(H')/4(2W-1) \\
&\geq \frac{(\vec{\pi}(G) - (W-1) - 2DP(2W-1))}{4(2W-1)}
\end{aligned}$$

since each node in $V(H')$ has weight at most $2W-1$ despite alterations on terminal weights and $\vec{\pi}'(H') \geq \vec{\pi}(G) - (W-1) - 2DP(2W-1)$. Therefore

$$\min\left(|U|, |V \setminus U|\right) \geq \frac{\vec{\pi}(G) - (W-1)}{4(2W-1)} - DP/2$$

□

Lemma 6.14. $|\delta_H(U)| \geq \phi(H)\left(\frac{K}{8} - DP/2\right)$, *for any* $(U, V(H) \setminus U)$ *as defined in Lemma 6.13.*

Proof. By the edge expansion property of expander H, we get the lower bound on the size of the cut $(U, V \setminus U)$:

$$\begin{aligned}
|\delta_H(U)| &\geq \phi(H)\min\left(|U|, |V \setminus U|\right) \\
&\geq \phi(H)\left(\frac{\vec{\pi}(G) - (W-1)}{4(2W-1)} - DP/2\right) \\
&\geq \phi(H)\left(\vec{\pi}(G)/8W + \vec{\pi}(G)/16W^2 - \frac{1}{8} - DP/2\right) \\
&\geq \phi(H)\left(\frac{K}{8} - DP/2\right)
\end{aligned}$$

since $K \leq \vec{\pi}(G)/W$, given that every cluster must have weight at least W in H and $\vec{\pi}(G)/16W^2 = \Omega(\log^3 n) \geq 1/8$. □

On the other hand, by Lemma 6.12, we know that the current size of the balanced cut in H' is at most the size of the multicut L given the construction of

$(U', V(H') \setminus U')$:

$$\left|\delta_{H'}(U')\right| \leq |E| \log n / D = \frac{K\omega \log^2 n \log n}{2D} = \frac{K\omega}{2a_3} \tag{6.5.3}$$

The edge loss from the balanced cut $\mathsf{cap}(U, V(H) \setminus U)$ in H is caused by routing the P paths, which can take away at most $DP\omega \log^2 n$ number of edges. Thus we have:

$$\begin{aligned}
\left|\delta_{H'}(U')\right| + DP\omega \log^2 n &\geq |\delta_H(U)| \\
\frac{\omega K}{2a_3} + DP\omega \log^2 n &\geq \phi(H)\left(\frac{K}{8} - DP/2\right) \\
DP(\omega \log^2 n + \phi(H)/2) &\geq \phi(H)\frac{K}{8} - \frac{K\omega}{2a_3} \\
P &\geq \frac{\left(\phi(H)\frac{K}{8} - \frac{K\omega}{2a_3}\right)}{a_3 \log^3 n(\omega \log^2 n + \phi(H)/2)}
\end{aligned}$$

By taking $\phi(H) = 1/4$, $a_3 = 32\omega$, we have $D = 32\omega \log^3 n$ and $P \geq K/2048\omega^2 \log^5 n$, for a constant ω. $\square$

Part III: Classification

Imagine – *John Lennon*

...

Imagine there's no countries
It isn't hard to do
Nothing to kill or die for
And no religion too
Imagine all the people
Living life in peace...

...

Imagine no possessions
I wonder if you can
No need for greed or hunger
A brotherhood of man
Imagine all the people
Sharing all the world...

7 Global and Local Optima Lemmas

Be careful what you wish for, it might come true. – J.K.Rowling

7.1 The Problem Definition and Preliminaries

In the next three chapters, we study the following problem: Given a set of $2N$ diploid individuals from two populations P_1 and P_2, we aim to classify individuals according to their populations of origin, based on only a small amount of their genotype data. Recall that for diploid organisms the choromosomes come in pairs and a genotype is a list of unordered pairs of alleles such that one comes from each parent. We define K as the number of attributes that we draw from each individual. We aim to minimize K while being able to classify our sample.

We use capital letters X, Y, Z to denote individuals, each of which also represents the observed genotype data corresponding to an individual across a set of K loci. Since each SNP has two variants (alleles), we use bit 1 and bit 0 to denote them. We use their corresponding lower case letters x^i, y^i, z^i to denote their bits at locus i. We use $X^i = \{x_a^i : a \in \{1,2\}\}$ to denote an unordered pair of bits (alleles) at locus i for individual X.

In Chapter 9, we consider the case that we are given only a single bit from each of K loci. This corresponds to the classic problem of learning mixtures of two product distributions over the K dimensional Boolean cube $\{0,1\}^K$, when attribute values across different dimensions are mutually independent.

7.1.1 The Statistical Model and Measure of Distance

Given the population of origin of each individual, the genotypes are assumed to be generated by drawing alleles independently from the appropriate population frequency distribution. We use $p_x^i = \mathbf{Pr}\left[x_a^i = 1\right]$ and $p_y^i = \mathbf{Pr}\left[y_a^i = 1\right], \forall i = 1, \dots, K$ to denote the "success" probability (frequency of an allele mapping to bit 1) at locus i in the population of origin of individual X and Y respectively.

In particular, assuming the population of origin for X is P_1 and Y comes from population P_2: we assume that $x_a^i, \forall a \in \{1,2\}, \forall i$ are independent Bernoulli random variables with success probability p_x^i, and $y_b^i, \forall b \in \{1,2\}, \forall i$ are independent Bernoulli random variables with success probability p_y^i, and we define

$$\begin{aligned}
\mathbf{Pr}\left[x_a^i = 1\right] &= p_1^i = p_x^i, && (7.1.1)\\
\mathbf{Pr}\left[x_a^i = 0\right] &= q_1^i = 1 - p_1^i, && (7.1.2)\\
\mathbf{Pr}\left[y_a^i = 1\right] &= p_2^i = p_y^i, && (7.1.3)\\
\mathbf{Pr}\left[y_a^i = 0\right] &= q_2^i = 1 - p_2^i. && (7.1.4)
\end{aligned}$$

For two populations P_1 and P_2, we call $\vec{p}_1, \vec{p}_2$ their centers, where

$$\begin{aligned}
\vec{p}_1 &= (p_1^1, p_1^2, \dots, p_1^K), && (7.1.5)\\
\vec{p}_2 &= (p_2^1, p_2^2, \dots, p_2^K). && (7.1.6)
\end{aligned}$$

We use the following γ to measure the *average* distance between P_1 and P_2 across their K loci (dimensions).

Definition 7.1. $\gamma(P_1, P_2) = \frac{\sum_{i=1}^K (p_1^i - p_2^i)^2}{K}$.

We use $X \sim \vec{p}_a, \forall a = 1,2$ to represent that X is a random node from population P_a. In the following definitions, we use X to represent the sequence of K unordered pairs of bits that we see from locus 1 to K on node X, where an *unordered pair of bits* refer to one of $\{00, 01, 10, 11\}$. Let $X^k, \forall k$ represent the pair of bits at locus k in X. Recall that an *indicator variable* is a discrete random variable that takes on only the value of 0 or 1.

Definition 7.2. *Given X, we define the following indicator random variables $I_{11}^k(X), I_{00}^k(X), I_{(01)}^k(X), \forall k = 1, \ldots, K$ such that:*

$$\begin{aligned} I_{11}^k(X) = 1 &\quad : \quad X^k = 11, \\ I_{00}^k(X) = 1 &\quad : \quad X^k = 00, \\ I_{(01)}^k(X) = 1 &\quad : \quad X^k = 01, or 10, \end{aligned}$$

where X^k denotes the unordered pair of bits observed at locus k in X.

Definition 7.3. *Given two bits x, y, $I_{x=y}$ indicates if $x = y$, i.e.,*

$$\begin{aligned} I_{x=y} = 1 &\quad : \quad x = y, \\ I_{x=y} = 0 &\quad : \quad x \neq y. \end{aligned}$$

Definition 7.4. *$\forall k \in [1, K]$, let us denote $f^k(X) = I_{00}^k(X) - I_{11}^k(X)$ such that*

$$f^k(X) = \begin{cases} -1 & : \quad I_{11}^k(X) = 1, \\ 0 & : \quad I_{(01)}^k(X) = 1, \\ +1 & : \quad I_{00}^k(X) = 1. \end{cases}$$

Finally, we introduce the following theorem and its corollary that we use throughout this thesis. We refer to both as Hoeffding bound.

Theorem 7.1. (Hoeffding [1963]) *If $X_1, X_2, \ldots, X_K$ are independent and $a_i \leq X_i \leq b_i, \forall i = 1, 2, \ldots, K$, and if $\bar{X} = (X_1 + \ldots + X_K)/K$ and $\mu = \mathbf{E}[\bar{X}]$, then for $t > 0$*

$$\mathbf{Pr}[\bar{X} - \mu \geq t] \leq e^{-2K^2t^2/\sum_{i=1}^K (b_i - a_i)^2}.$$

We also use the following bound for the distribution function of the difference of two sample means.

Corollary 7.1. (Hoeffding [1963]) *If $Y_1, \ldots, Y_n$, $Z_1, \ldots, Z_m$ are independent random variables with values in the interval $[a, b]$, and if $\bar{Y} = (Y_1 + \ldots + Y_m)/m$, $\bar{Z} = (Z_1 + \ldots + Z_n)/n$, then for $t > 0$*

$$\mathbf{Pr}[\bar{Y} - \bar{Z} - (\mathbf{E}[\bar{Y}] - \mathbf{E}[\bar{Z}]) \geq t] \leq e^{-2t^2/(m^{-1}+n^{-1})(b-a)^2}.$$

7.2 Every Edge Has a Perfect Score

In this section, we prove a theorem regarding a score that we assign to each pair of individuals based on their genotype data. Given a large enough K, in particular, $K \geq \Omega(\ln N/\gamma^2)$, we show that scores between people from the same population are consistently lower than scores between people from different populations. Using this score, we can construct a complete graph where nodes are individuals and edge weight is the score between two individuals.

In particular, we call this score $\mathsf{Pscore}(X,Y)$. For an unordered pair of individuals (X,Y),

Definition 7.5.

$$\mathit{Pscore}(X,Y) = \sum_{i=1}^{K} \mathit{Pscore}^i(X,Y) = \sum_{i=1}^{K} \mathit{Pscore}^i \begin{bmatrix} x_1^i & x_2^i \\ y_1^i & y_2^i \end{bmatrix},$$

where

$$\mathit{Pscore}^i(X,Y) = \frac{1}{2}\begin{pmatrix} (I_{x_1^i=x_2^i} + I_{y_1^i=y_2^i}) - (I_{x_1^i=y_1^i} + I_{x_2^i=y_2^i}) + \\ (I_{x_1^i=x_2^i} + I_{y_1^i=y_2^i}) - (I_{x_1^i=y_2^i} + I_{x_2^i=y_1^i}) \end{pmatrix}.$$

Note that this definition utilizes a special quartet construction involving four bits $x_1^i, x_2^i, y_1^i, y_2^i$ that are four independent Bernoulli random variables such that two bits from each pair (x_1^i, x_2^i), (y_1^i, y_2^i) are identically distributed.

Table 7.2 shows the scores, where the top row denotes $(x_1^i x_2^i)$ while the first column denotes $(y_1^i y_2^i)$; we only define scores for 01 since 01 and 10 are equivalent as an unordered pair to $\mathsf{Pscore}^i(X,Y)$.

	00	11	01
00	0	2	0
11	2	0	0
01	0	0	-1

Table 7.1. A TABLE ILLUSTRATING $\mathsf{PSCORE}^i, \forall i$ GIVEN ANY FOUR BITS

Lemma 7.1. *If X,Y come from different populations and Z_1,Z_2 are of common origin, then,*

$$
\begin{aligned}
\mathbf{E}[\mathit{Pscore}(X,Y)] &= 2K\gamma, \\
\mathbf{E}[\mathit{Pscore}(Z_1,Z_2)] &= 0.
\end{aligned}
$$

Proof. We first define $\delta_x^i = (p_x^i)^2 + (q_x^i)^2$ and $\delta_{xy}^i = p_x^i p_y^i + q_x^i q_y^i$, where $q_x^i = 1 - p_x^i$ and $q_y^i = 1 - p_y^i$, and hence

$$
\begin{aligned}
\mathbf{E}\left[I_{x_1^i=x_2^i} + I_{y_1^i=y_2^i} - (I_{x_1^i=y_1^i} + I_{x_2^i=y_2^i})\right] &= \mathbf{E}\left[I_{x_1^i=x_2^i} + I_{y_1^i=y_2^i} - (I_{x_1^i=y_2^i} + I_{x_2^i=y_1^i})\right] \\
&= \delta_x^i + \delta_y^i - 2\delta_{xy}^i \\
&= 2(p_x^i - p_y^i)^2. \qquad (7.2.7)
\end{aligned}
$$

Given that X,Y come from different populations, we apply (7.2.7),

$$
\begin{aligned}
\mathbf{E}[\mathsf{Pscore}(X,Y)] &= \sum_{i=1}^{K} \mathbf{E}\left[\mathsf{Pscore}^i(X,Y)\right] \qquad (7.2.8) \\
&= \frac{1}{2}\sum_{i=1}^{K} \mathbf{E}\left[I_{x_1^i=x_2^i} + I_{y_1^i=y_2^i} - (I_{x_1^i=y_1^i} + I_{x_2^i=y_2^i})\right] + \\
&\quad \frac{1}{2}\sum_{i=1}^{K} \mathbf{E}\left[I_{x_1^i=x_2^i} + I_{y_1^i=y_2^i} - (I_{x_1^i=y_2^i} + I_{x_2^i=y_1^i})\right] \qquad (7.2.9) \\
&= 2\sum_{i=1}^{K} (p_1^i - p_2^i)^2 = 2K\gamma. \qquad (7.2.10)
\end{aligned}
$$

For Z_1,Z_2, $\forall i$, $p_{Z_1}^i = p_{Z_2}^i$, hence $\mathbf{E}[\mathsf{Pscore}(Z_1,Z_2)] = 0$. ☐

The following theorem does not assume that the sample contains the same number of points from each population.

Theorem 7.2. *Let the sample size be $2N$. Given that $K \geq 18\ln N/\gamma^2$, with probability $1 - O(1/N^2)$, for any four individuals X,Y,Z_1,Z_2 such that X,Y come from different populations and Z_1,Z_2 come from the same population,*

$$\mathit{Pscore}(X,Y) \geq \mathit{Pscore}(Z_1,Z_2).$$

Proof. We first use Hoeffding bound to prove the following lemma.

Lemma 7.2. *Given that* $K \geq 18 \ln N/\gamma^2$,

$$\begin{aligned} \mathbf{Pr}[\mathit{Pscore}(Z_1, Z_2) \geq K\gamma] &< 1/N^4, \\ \mathbf{Pr}[\mathit{Pscore}(X, Y) \leq K\gamma] &< 1/N^4. \end{aligned}$$

Proof. Given that $\mathbf{E}[\mathsf{Pscore}(Z_1, Z_2)] = 0$ and $\mathsf{Pscore}(Z_1, Z_2) = \sum_{i=1}^{K} \mathsf{Pscore}^i(Z_1, Z_2)$ is the sum of K independent random variables with values in $[-1, 2]$, using Hoeffding bound as in Theorem 7.1 with $t = K\gamma/K = \gamma$,

$$\mathbf{Pr}[\mathsf{Pscore}(Z_1, Z_2) \geq K\gamma] = e^{-2K^2(\gamma)^2/K(3)^2} \leq 1/N^4. \tag{7.2.11}$$

Similarly, given that $\mathbf{E}[\mathsf{Pscore}(X, Y)] = 2K\gamma$,

$$\begin{aligned} \mathbf{Pr}[\mathsf{Pscore}(X, Y) \leq K\gamma] = \\ \mathbf{Pr}[-\mathsf{Pscore}(X, Y) + \mathbf{E}[\mathsf{Pscore}(X, Y)] \geq K\gamma] \quad & (7.2.12) \\ = \quad e^{-2K^2(\gamma)^2/K(3)^2} \leq 1/N^4. \quad & (7.2.13) \end{aligned}$$

□

By union bound, the probability that any event of type $\mathsf{Pscore}(X, Y) \leq K\gamma$ or type $\mathsf{Pscore}(Z_1, Z_2) \geq K\gamma$ happens is at most $4N^2/N^4$, since the total number of such events are $2N(2N - 1)$. Hence the theorem holds. □

7.3 How to Learn Which Side to Join?

In this section, we study the following problem: Assume that we have separated $2N$ individuals, N from each population of origin, and we are now given a new node X that we need to place on the *correct* side according to its population of origin. First recall that we use X to represent the individual X's genotype data over its K loci. Hence $X = \{\{x_1^k, x_2^k\}, \forall k\}$ is a sequence of K pairs of unordered bits, which we also refer to as the bit string of X loosely.

We prove the Local Optimum Lemma as follows. Given a fixed individual X with a certain bit string, and its N peers from each population: $X_1, \ldots, X_N$ and

$Y_1, \ldots, Y_N$, we show that given a large enough K, i.e., given enough loci, which is parameterized over N and γ, with high probability,

$$\sum_{i=1}^{N} \mathsf{Pscore}(X, Y_i) > \sum_{i=1}^{N} \mathsf{Pscore}(X, X_i). \tag{7.3.14}$$

Thus we can put X on its own population side given that all of its $2N$ peers $X_1, \ldots, X_N, Y_1, \ldots, Y_N$ have been placed correctly.

Fix X to be $\tilde{X}$. Let $\mathsf{Pscore}(\tilde{X}, Z)$ denote $\mathsf{Pscore}(X, Z | X = \tilde{X})$. The idea of the proof is that while $\mathsf{Pscore}(X, X_i), \mathsf{Pscore}(X, Y_i), \forall i$ are all dependent on random variable X, they become mutually independent once we fix X to an arbitrary bit string $\tilde{X}$ that we could possibly observe. In other words, random variables $\mathsf{Pscore}(\tilde{X}, Y_i), \forall i = 1, \ldots, N$, and $\mathsf{Pscore}(\tilde{X}, X_i), \forall i = 1, \ldots, N$ are conditionally independent given X being fixed to $\tilde{X}$. This allows us to use Hoeffding bound to prove the Local Optimum Lemma as in Theorem 7.3.

We define the following random variable $\mathsf{diff}(X)$ such that $\mathsf{diff}(\tilde{X})$ represents the expected difference of two conditionally independent random variables $\mathsf{Pscore}(\tilde{X}, Y_i)$, $\mathsf{Pscore}(\tilde{X}, X_i)$, each of which depends on either Y_i or X_i given that X is fixed to $\tilde{X}$. Hence expectations are taken over all possible realizations of Y_i, X_i respectively.

Definition 7.6. *Let $X \sim \vec{p}_1$ be a node from P_1 and $Y \sim \vec{p}_2$ be a node from P_2. Let $Z_1 \sim \vec{p}_1$, $Z_2 \sim \vec{p}_2$ be two nodes randomly drawn from P_1 and P_2 respectively.*

$$\begin{aligned} \mathit{diff}(X) &= \mathbf{E}_{Z_2 \sim \vec{p}_2}[\mathit{Pscore}(X, Z_2)] - \mathbf{E}_{Z_1 \sim \vec{p}_1}[\mathit{Pscore}(X, Z_1)], \\ \mathit{diff}(Y) &= \mathbf{E}_{Z_1 \sim \vec{p}_1}[\mathit{Pscore}(Y, Z_1)] - \mathbf{E}_{Z_2 \sim \vec{p}_2}[\mathit{Pscore}(Y, Z_2)]. \end{aligned}$$

Remark 7.1. *Hence $\mathit{diff}(X)$ is determined by node X's bit string $\tilde{X}$. $\mathit{diff}(X)$ is a random variable that is completely determined by the outcome $\tilde{X}$ of X. For a fixed outcome $\tilde{X}$, $\mathit{diff}(X)$ is a fixed value.*

Proposition 7.1. $\forall a = 1, 2$, $\mathbf{E}_{X \sim \vec{p}_a}[\mathit{diff}(X)] = 2K\gamma$.

Proof. W.l.o.g., assume that $X \sim \vec{p}_1$; for $X \sim \vec{p}_2$, proof is similar.

$$\begin{aligned}
&\mathbf{E}_{X\sim\vec{p}_1}[\mathsf{diff}(X)] = \\
&\qquad \mathbf{E}_{X\sim\vec{p}_1}\left[\mathbf{E}_{Z_2\sim\vec{p}_2}[\mathsf{Pscore}(X,Z_2)] - \mathbf{E}_{Z_1\sim\vec{p}_1}[\mathsf{Pscore}(X,Z_1)]\right] \\
&= \quad \mathbf{E}_{X\sim\vec{p}_1,Z_2\sim\vec{p}_2}[\mathsf{Pscore}(X,Z_2)] - \mathbf{E}_{X\sim\vec{p}_1,Z_1\sim\vec{p}_1}[\mathsf{Pscore}(X,Z_1)] \\
&= \quad 2K\gamma,
\end{aligned}$$

where the last step is due to Lemma 7.1. □

We show that when K is big enough, which is parameterized over N and γ, with high probability, the observed bit string $\tilde{X}$ is conforming to what we expect to see; specifically, this is reflected in the bounded deviation of $\mathsf{diff}(X)$ from its expected value $2K\gamma$, given $\tilde{X}$. Indeed,

Lemma 7.3. *Given that* $K \geq \frac{8\ln 1/\tau}{\gamma}$, $\mathbf{Pr}[\mathit{diff}(X) > K\gamma] > 1-\tau$.

Proof. Given $\tilde{X}$, we use the following indicator random variables $I_{11}^k(X), I_{00}^k(X), I_{(01)}^k(X), \forall k = 1,\ldots,K$ as defined in Definition 7.2, where $X^K = \widetilde{x_1^k x_2^k}$ denotes the unordered pair of bits observed at locus K in $\tilde{X}$.

We first show the following claim.

Claim 7.1. *Let X be any sample point.* $\forall a = 1,2$,

$$\begin{aligned}
&\mathbf{E}_{Y\sim\vec{p}_a}\left[\mathit{Pscore}(X,Y|X=\tilde{X})\right] = \\
&\qquad 2\sum_{k=1}^{K} I_{11}^k(X)(q_a^k)^2 + 2\sum_{k=1}^{K} I_{00}^k(X)(p_a^k)^2 - 2\sum_{k=1}^{K} I_{(01)}^k(X)\,p_a^k q_a^k.
\end{aligned}$$

Proof. It is straightforward to verify that for each locus i,

$$\mathsf{Pscore}^i(\tilde{X},Y) = \begin{cases}
2 & : \begin{pmatrix} \tilde{x}_1^i & \tilde{x}_2^i \\ y_1^i & y_2^i \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}, \text{ or } \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}, \\
-1 & : \begin{pmatrix} \tilde{x}_1^i & \tilde{x}_2^i \\ y_1^i & y_2^i \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \text{ or } \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \\
 & : \text{ or } \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \text{ or } \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \\
0 & : \text{ otherwise.}
\end{cases}$$

Therefore we have

$$
\begin{aligned}
& \mathbf{E}_{Y\sim\vec{p}_a}\left[\mathsf{Pscore}(X,Y|X=\tilde{X})\right] \\
= & \sum_{k=1}^{K}\mathbf{E}_{Y\sim\vec{p}_a}\left[\mathsf{Pscore}^k(X,Y|X=\tilde{X})\right] \\
= & 2\sum_{k=1}^{K}(I_{11}^k(X)(q_a^k)^2+I_{00}^k(X)(p_a^k)^2-\frac{1}{2}I_{(01)}^k(X)2p_a^kq_a^k) \\
= & 2\sum_{k=1}^{K}I_{11}^k(X)(q_a^k)^2+2\sum_{k=1}^{K}I_{00}^k(X)(p_a^k)^2-2\sum_{k=1}^{K}I_{(01)}^k(X)p_a^kq_a^k.
\end{aligned}
$$

□

We next use Claim 7.1, and function $f^k(X)=I_{00}^k(X)-I_{11}^k(X)$, as specified in Definition 7.4, to derive (7.3.15) for $X\sim\vec{p}_1$, and (7.3.16) for $Y\sim\vec{p}_2$, where

$$
\begin{aligned}
\rho_k &= (p_2^k)^2-(p_1^k)^2, & \omega_k &= (p_2^kq_2^k-p_1^kq_2^k), \\
\psi_k &= (q_2^k)^2-(q_1^k)^2, & S &= 2\sum_{k=1}^{K}(p_2^k-p_1^k)(p_2^k+p_1^k-1),
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{diff}(X) &= \mathbf{E}_{Z_2\sim\vec{p}_2}\left[\mathsf{Pscore}(X,Z_2|X=\tilde{X})\right]-\mathbf{E}_{Z_1\sim\vec{p}_1}\left[\mathsf{Pscore}(X,Z_1|X=\tilde{X})\right] \\
&= 2\sum_{k=1}^{K}I_{11}^k(X)\psi^k+2\sum_{k=1}^{K}I_{00}^k(X)\rho^k-2\sum_{k=1}^{K}I_{(01)}^k(X)\omega^k) \\
&= 2\sum_{k=1}^{K}(p_2^k-p_1^k)(I_{00}^k(X)-I_{11}^k(X))+2\sum_{k=1}^{K}(p_2^k-p_1^k)(p_2^k+p_1^k-1) \\
&= 2\sum_{k=1}^{K}(p_2^k-p_1^k)f^k(X)+S, \qquad (7.3.15) \\
\mathsf{diff}(Y) &= \mathbf{E}_{Z_1\sim\vec{p}_1}\left[\mathsf{Pscore}(Y,Z_1|Y=\tilde{Y})\right]-\mathbf{E}_{Z_2\sim\vec{p}_2}\left[\mathsf{Pscore}(Y,Z_2|Y=\tilde{Y})\right] \\
&= 2\sum_{k=1}^{K}(p_1^k-p_2^k)(I_{00}^k(Y)-I_{11}^k(Y))+2\sum_{k=1}^{K}(p_1^k-p_2^k)(p_2^k+p_1^k-1) \\
&= 2\sum_{k=1}^{K}(p_1^k-p_2^k)f^k(Y)-S. \qquad (7.3.16)
\end{aligned}
$$

Let us define μ_f, and by proposition 7.1, (7.3.15) and (7.3.16),

$$\begin{aligned}
\mathbf{E}_{X\sim\vec{p}_1}\left[2\sum_{k=1}^{K}(p_2^k - p_1^k)f^k(X)\right] &= \mu_f && (7.3.17)\\
&= \mathbf{E}_{X\sim\vec{p}_1}[\mathsf{diff}(X)] - S\\
&= 2K\gamma - S, && (7.3.18)\\
\mathbf{E}_{Y\sim\vec{p}_2}\left[2\sum_{k=1}^{K}(p_1^k - p_2^k)f^k(Y)\right] &= \mathbf{E}_{Y\sim\vec{p}_2}[\mathsf{diff}(Y)] + S\\
&= 2K\gamma + S. && (7.3.19)
\end{aligned}$$

We now show the following claims using Hoeffding bound; we only show proof for Claim 7.2, since proof for the other one is similar.

Claim 7.2. *Given that* $K \geq \frac{8\ln 1/\tau}{\gamma}$,

$$\mathbf{Pr}_{X\sim\vec{p}_1}\left[\sum_{k=1}^{K}2(p_2^k - p_1^k)f^k(X) - \mu_f \leq -K\gamma\right] \leq \tau.$$

Claim 7.3. *Given that* $K \geq \frac{8\ln 1/\tau}{\gamma}$,

$$\mathbf{Pr}_{Y\sim\vec{p}_2}\left[\sum_{k=1}^{K}2(p_1^k - p_2^k)f^k(Y) - (2K\gamma + S) \leq -K\gamma\right] \leq \tau.$$

Proof of claim 7.2: Given that each observed bit in $\tilde{X}$ is an independent Bernoulli random variable, $f^1(X), f^2(X), \ldots, f^K(X)$ are independent and $-2\sqrt{\gamma_k} \leq 2(p_2^k - p_1^k)f^k(X) \leq 2\sqrt{\gamma_k}$, where $\gamma_k = (p_2^k - p_1^k)^2, \forall k = 1,\ldots,K$, then for $t = K\gamma/K = \gamma$,

$$\mathbf{Pr}_{X\sim\vec{p}_1}\left[\sum_{k=1}^{K}2(p_2^k - p_1^k)f^k(X) - \mu_f \leq -K\gamma\right] \quad \leq \quad e^{-2K^2(\gamma)^2/\sum_{k=1}^{K}(4\sqrt{\gamma_k})^2} \leq \tau.$$

□

These claims immediately imply Lemma 7.3, since

$$\mathbf{Pr}_{X\sim\vec{p}_1}\left[\sum_{k=1}^{K}\mathsf{diff}(X)-2K\gamma\leq-K\gamma\right]=$$
$$\mathbf{Pr}_{X\sim\vec{p}_1}\left[\sum_{k=1}^{K}2(p_2^k-p_1^k)f^k(X)-\mu_f\leq-K\gamma\right]\leq\tau,$$
$$\mathbf{Pr}_{Y\sim\vec{p}_2}\left[\sum_{k=1}^{K}\mathsf{diff}(Y)-2K\gamma\leq-K\gamma\right]=$$
$$\mathbf{Pr}_{Y\sim\vec{p}_2}\left[\sum_{k=1}^{K}2(p_1^k-p_2^k)f^k(Y)-(2K\gamma+S)\leq-K\gamma\right]\leq\tau.$$

□

W.l.o.g., we assume that $X\sim\vec{p}_1$. We have shown that there is a significant difference in the expected values, given enough multilocus genotype data (e.g., SNPs); that is, with high probability, we will observe a bit string $\tilde{X}$ of node $X\sim\vec{p}_1$ such that $\forall i=1,\ldots,N$,

$$\mathbf{E}_{Y_i\sim\vec{p}_2}\left[\mathsf{Pscore}(X,Y_i|X=\tilde{X})\right]-\mathbf{E}_{X_i\sim\vec{p}_1}\left[\mathsf{Pscore}(X,X_i|X=\tilde{X})\right]>K\gamma, \quad (7.3.20)$$

due to the bounded amount of deviation in random variable $\mathsf{diff}(X)$ from its expected value, when evaluated at $\tilde{X}$. A similar statement holds for $Y\sim\vec{p}_2$.

We are ready to show the theorem of this section. The theorem shows that with high probability, we can place a node X in the correct side given enough number of loci and N random peers from each population. In particular, the failure probability comes from either X being a *bad node* such that $\mathsf{diff}(X)\leq K\gamma$, or from a certain large deviation event for the sum of $2KN$ conditional independent random variables as shown in the proof.

Theorem 7.3. *Let* $K\geq\max\{\frac{9\ln(1/\delta)}{N\gamma^2},\frac{8\ln(1/\tau)}{\gamma}\}$. *For any* $X\sim\vec{p}_1$ *and its observed bit string* $\tilde{X}$, *and* $2N$ *individuals* $X_i\sim\vec{p}_1,Y_i\sim\vec{p}_2,\forall i=1,\ldots,N$ *that are randomly* *drawn from their populations of origin, with probability* $1-\tau-\delta$,

$$\sum_{i=1}^{N} \mathit{Pscore}(X,Y_i|X=\tilde{X}) > \sum_{i=1}^{N} \mathit{Pscore}(X,X_i|X=\tilde{X}).$$

A similar statement holds for $Y \sim \vec{p}_2$.

Proof. Given an individual X and its observed bit string $\tilde{X}$, we first define $2NK$ conditional independent random variables, such that $\forall i = 1,\ldots,N$, $\mathsf{Pscore}(Y_i,X|X=\tilde{X})$ and $\mathsf{Pscore}(X_i,X|X=\tilde{X})$ each contribute K conditional independent random variables that are also conditional independent with respect to all other $(2N-1)K$ such random variables.

Let $Y_i^k = \mathsf{Pscore}^k(X,Y_i|X=\tilde{X})$ and $Z_i^k = \mathsf{Pscore}^k(X,X_i|X=\tilde{X})$ be the $2NK$ conditional independent random variables with values in $[-1,2]$, and

$$\bar{Y} = \sum_{i=1}^{N}\sum_{k=1}^{K} \mathsf{Pscore}^k(X,Y_i|X=\tilde{X})/NK, \tag{7.3.21}$$

$$\bar{Z} = \sum_{i=1}^{N}\sum_{k=1}^{K} \mathsf{Pscore}^k(X,X_i|X=\tilde{X})/NK. \tag{7.3.22}$$

Claim 7.4. *Given that* $K \geq \frac{8\ln(1/\tau)}{\gamma}$ *and a particular bit string* $\tilde{X}$ *for node* $X \sim \vec{p}_1$, *with probability* $1-\tau$,

$$\mathbf{E}[\bar{Y}] - \mathbf{E}[\bar{Z}] > \gamma.$$

Proof. Using Lemma 7.3, we have $\mathsf{diff}(X) > K\gamma$, with probability $1-\tau$, given that $K \geq \frac{8\ln(1/\tau)}{\gamma}$.

Hence given that individuals $X_i \sim \vec{p}_1, Y_i \sim \vec{p}_2, \forall i = 1,\ldots,N$, are randomly drawn from their populations of origin, we have with probability $1-\tau$,

$$\begin{aligned}
\mathbf{E}[\bar{Y}] - \mathbf{E}[\bar{Z}] &= \mathbf{E}[(\bar{Y}-\bar{Z})] \\
&= \frac{1}{NK}\sum_{i=1}^{N} \mathbf{E}\left[(\mathsf{Pscore}(X,Y_i|X=\tilde{X}) - \mathsf{Pscore}(X,X_i|X=\tilde{X}))\right] \\
&= \frac{1}{NK}\sum_{i=1}^{N} \mathsf{diff}(X) > NK\gamma/NK = \gamma.
\end{aligned}$$

□

Assuming that we indeed have observed a bit string $\tilde{X}$ such that $\mathbf{E}[\bar{Y}] - \mathbf{E}[\bar{Z}] > \gamma$, we apply Corollary 7.1 of Theorem 7.1; Given that $t = \mathbf{E}[\bar{Y}] - \mathbf{E}[\bar{Z}] > \gamma$ and $K \geq \frac{9\ln(1/\delta)}{N\gamma^2}$,

$$\begin{aligned}
& \mathbf{Pr}\left[\sum_{i=1}^{N} \mathsf{Pscore}(X, Y_i | X = \tilde{X}) - \sum_{i=1}^{N} \mathsf{Pscore}(X, X_i | X = \tilde{X}) < 0 | \mathbf{E}[\bar{Y}] - \mathbf{E}[\bar{Z}] \geq \gamma\right] \\
= \;& \mathbf{Pr}[\bar{Y} - \bar{Z} - (\mathbf{E}[\bar{Y}] - \mathbf{E}[\bar{Z}]) \leq -(\mathbf{E}[\bar{Y}] - \mathbf{E}[\bar{Z}])] \\
= \;& \mathbf{Pr}[-(\bar{Y} - \bar{Z}) + (\mathbf{E}[\bar{Y}] - \mathbf{E}[\bar{Z}]) \geq (\mathbf{E}[\bar{Y}] - \mathbf{E}[\bar{Z}])] \\
\leq \;& e^{\frac{-2(t)^2}{2(3)^2/NK}} \leq e^{\frac{-2(\gamma)^2}{2(3)^2/NK}} \\
\leq \;& \delta.
\end{aligned}$$

Thus the total probability of a bad event

$$\begin{aligned}
& \mathbf{Pr}\left[\sum_{i=1}^{N} \mathsf{Pscore}(X, Y_i | X = \tilde{X}) - \sum_{i=1}^{N} \mathsf{Pscore}(X, X_i | X = \tilde{X}) < 0\right] = \\
& \quad \mathbf{Pr}[\mathbf{E}[\bar{Y}] - \mathbf{E}[\bar{Z}] \leq \gamma] + \\
& \quad \mathbf{Pr}\left[\sum_{i=1}^{N} \mathsf{Pscore}(X, Y_i | X = \tilde{X}) - \sum_{i=1}^{N} \mathsf{Pscore}(X, X_i | X = \tilde{X}) < 0 | \mathbf{E}[\bar{Y}] - \mathbf{E}[\bar{Z}] \geq \gamma\right] \\
\leq \;& \tau + \delta.
\end{aligned}$$

□

Corollary 7.2. *Let $K \geq \max\{\frac{18\ln N}{N\gamma^2}, \frac{16\ln N}{\gamma}\}$. For any $X \sim \vec{p}_1$ and its observed string $\tilde{X}$, with probability $1 - O(1/N^2)$, given that $X_i \sim \vec{p}_1, Y_i \sim \vec{p}_2, \forall i$ are individuals randomly drawn from their population of origin, we have*

$$\sum_{i=1}^{N} \mathit{Pscore}(X, Y_i | X = \tilde{X}) > \sum_{i=1}^{N} \mathit{Pscore}(X, X_i | X = \tilde{X}).$$

A similar statement holds for $Y \sim \vec{p}_2$.

8 Recognizing a Perfect Partition

8.1 Introduction

In this chapter, we aim to classify a balanced input instance and derive a tighter bound on K based on the Pscore that we define in section 7.2 and the complete graph that we construct, where nodes are individuals and edge weight is the score between two individuals. Let P_1 represent the set of nodes $X_1, X_2, \ldots, X_N$ from population 1, and P_2 represent the set of nodes $Y_1, Y_2, \ldots, Y_N$ from population 2. Recall that a cut $(S, \bar{S})$ refers to the set of edges with exactly one endpoint in S. We define Pscore for a cut $(S, \bar{S})$ as the sum of Pscores over the set of edges in $(S, \bar{S})$. When we say Pscore over an edge $e = (u, v)$, we refer to $\mathsf{Pscore}(u, v)$.

Consider a balanced cut $(S, \bar{S})$, as shown in Figure 8.1.1, where

$$S = \{X_i \in P_1, i = 1, \ldots, N-L, V_j \in P_2, j = 1, \ldots, L\}, \tag{8.1.1}$$

$$\bar{S} = \{Y_i \in P_2, i = 1, \ldots, N-L, U_j \in P_1, j = 1, \ldots, L\}, \tag{8.1.2}$$

and $L \in [1, N/2]$ is the number of nodes that have been swapped from one side of $\mathcal{T}$ to the other, by definition,

$$\begin{aligned} \mathsf{Pscore}(S, \bar{S}) = & \sum_{i=1}^{N-L} \sum_{j=1}^{N-L} \mathsf{Pscore}(X_i, Y_j) + \sum_{i=1}^{L} \sum_{j=1}^{L} \mathsf{Pscore}(U_i, V_j) + \\ & \sum_{i=1}^{N-L} \sum_{j=1}^{L} (\mathsf{Pscore}(X_i, U_j) + \mathsf{Pscore}(Y_i, V_j)), \end{aligned} \tag{8.1.3}$$

which defines $\mathsf{Pscore}(\mathcal{T})$ when $L = 0$, i.e.,

$$\mathsf{Pscore}(\mathcal{T}) = \sum_{i=1}^{N}\sum_{j=1}^{N} \mathsf{Pscore}(X_i, Y_j). \tag{8.1.4}$$

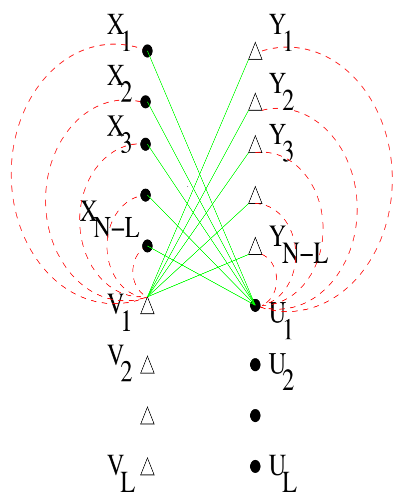

Figure 8.1.1. Edges that are different between a perfect partition $\mathcal{T}$ and another balanced partition $(S,\bar{S})$, seen only from $U_1 \sim \vec{p}_1$ and $V_1 \sim \vec{p}_2$; red dotted edges are in $\mathcal{T}$ and green solid edges are in $(S,\bar{S})$.

It is easy to verify that in expectation, the perfect partition has the maximum Pscore, i.e., $\forall$ balanced $(S,\bar{S})$ other than $\mathcal{T}$, $\mathbf{E}[\mathsf{Pscore}(\mathcal{T})] > \mathbf{E}\left[\mathsf{Pscore}(S,\bar{S})\right]$. Furthermore, the following theorem says that this is also true with high probability, given a large enough K. Formally,

Theorem 8.1. *Given that $K = \Omega(\frac{\ln N}{\gamma})$ and $KN = \Omega(\frac{\ln N \log\log N}{\gamma^2})$, where $N \geq 8$, with probability $1 - 1/\operatorname{poly}(N)$, for all other balanced cut $(S,\bar{S})$ in the complete graph formed among $2N$ nodes, we have*

$$\mathit{Pscore}(\mathcal{T}) > \mathit{Pscore}(S,\bar{S}).$$

Remark 8.1. *When $N = \Omega(\log\log N/\gamma)$, i.e., when we have enough individuals from each population, $K = \Omega(\frac{\ln N}{\gamma})$ becomes the only constraint.*

8.1.1 The Approach

We compare each balanced cut $(S,\bar{S})$ against the perfect partition $\mathcal{T}$, and define a random variable $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)$ as in (8.1.5) to capture their difference. Figure 8.1.1 shows the nodes that we refer to in (8.1.6),

$$\begin{aligned}
\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) &= \mathsf{Pscore}(\mathcal{T}) - \mathsf{Pscore}(S,\bar{S}) && (8.1.5)\\
&= \sum_{j=1}^{L}\sum_{i=1}^{N-L}\mathsf{Pscore}(V_j,X_i) - \mathsf{Pscore}(V_j,Y_i) + \\
&\quad \sum_{j=1}^{L}\sum_{i=1}^{N-L}\mathsf{Pscore}(U_j,Y_i) - \mathsf{Pscore}(U_j,X_i). && (8.1.6)
\end{aligned}$$

Thus for a particular balanced cut $(S,\bar{S})$, $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) > 0$ immediately implies that $\mathsf{Pscore}(\mathcal{T}) > \mathsf{Pscore}(S,\bar{S})$. And this is what Theorem 8.1 aims to prove for all balanced cuts.

In more detail, we refer to $X_i \in (S \cap P_1)$ and $Y_i \in (\bar{S} \cap P_2)$, $\forall i \in [1,N-L]$ as *unswapped* nodes, since they belong to the majority type in their own side; we denote $V_j \in (S \cap P_2), U_j \in (\bar{S} \cap P_1), \forall j \in [1,L]$ as *swapped* nodes since they are the minority on the their new side.

The random variable $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L), \forall N/2 \geq L \geq 1$, comprises exactly of Pscores over the set of edges that differ between those in $\mathcal{T}$ and those in $(S,\bar{S})$, which is exactly the set of $4L(N-L)$ edges between swapped nodes and unswapped nodes, among which $4(N-L)$ edges are shown in Figure 8.1.1.

In particular, for $(S,\bar{S})$, as shown in Figure 8.1.1, original cut edges $\{(V_j,X_i),(U_j,Y_i), \forall j \in [1,L], \forall i \in [1,N-L]\}$ that belong to $\mathcal{T}$ are replaced with $\{(V_j,Y_i),(U_j,X_i), \forall j \in [1,L], \forall i \in [1,N-L]\}$, which are the *new edges* that appear in $(S,\bar{S})$; these *new edges* together with the set of common edges that belong to $\mathcal{T} \cap (S,\bar{S})$ form $(S,\bar{S})$. Hence we only need to consider the influence of $2NK$ random pairs of bits over these two sets of edges, as shown in (8.1.6), $\forall (S,\bar{S})$.

In particular, observe that all random variables, $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L), \forall (S,\bar{S})$, $\forall L > 0$ have positive expected values, as in Proposition 8.1, as their initial *advantage*; thus we need to show that the deviation of each random variable from its expected value is less than the expected advantage with high probability.

Proposition 8.1. $\mathbf{E}\left[\mathit{diff}(\mathcal{T},(S,\overline{S}),L)\right] = 4L(N-L)K\gamma$, *where expectation is over all K random pairs of bits of $X_i, Y_i, \forall i \in [1, N-L]$ and $U_j, V_j, \forall j \in [1, L]$.*

We include in the next section some preliminaries on probability theory due to our intensive use of these terms in this chapter.

8.2 Preliminaries On Probability Theory

Most of the following definitions come from the textbook Randomized Algorithms by Motwani and Raghavan [1995]. Some others come from a paper by Chung and Lu [2006].

In all definitions below, we shall be thinking of some sample space Ω, and when we speak of the complement of A, denoted as A^c, we mean all those elements of Ω which are not the elements of A.

First recall that a σ-field is the following.

Definition 8.1. *A σ-field $(\Omega, \mathcal{F})$ consists of a sample space Ω and a collection of subsets of Ω, denoted as $\mathcal{F}$, satisfying the following conditions.*

- $\emptyset \in \mathcal{F}$.
- *If $A \in \mathcal{F}$, then $A^c \in \mathcal{F}$.*
- *If $A_1, A_2, \ldots$ is a sequence of elements of $\mathcal{F}$ then*

$$\bigcup_j A_j \in \mathcal{F}$$

Definition 8.2. *Given a σ-field $(\Omega, \mathcal{F})$, a probability measure $\mathbf{Pr} : \mathcal{F} \to \mathbb{R}^+$ is a function that satisfies the following conditions.*

- $\forall A \in \mathcal{F}, 0 \le \mathbf{Pr}[A] \le 1$.
- $\mathbf{Pr}[\Omega] = 1$.
- *For mutually disjoint events $\varepsilon_1, \varepsilon_2, \ldots, \mathbf{Pr}[\cup_i \varepsilon_i] = \sum_i \mathbf{Pr}[\varepsilon_i]$.*

Definition 8.3. *A probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$ consists of a σ-field $(\Omega, \mathcal{F})$ with a probability measure $\mathbf{Pr}$ defined on it.*

Definition 8.4. *Given the σ-field $(\Omega, \mathcal{F})$ with $\mathcal{F} = 2^{\Omega}$, a filter* $\mathbf{F}$ *is nested sequence $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \ldots \subseteq \mathcal{F}_n$ of subsets of 2^{Ω} such that*

- $\mathcal{F}_0 = \{\emptyset, \Omega\}$
- $\mathcal{F}_n = 2^{\Omega}$
- *for $0 \leq i \leq n$, $(\Omega, \mathcal{F}_i)$ is a σ-field.*

Definition 8.5. *If $\varepsilon_1, \varepsilon_2, \ldots$ are disjoint events that partition Ω, then an event is in the generated σ-field $\mathcal{F}$ if and only if it can be expressed as a union of some subset of the events $\varepsilon_1, \varepsilon_2, \ldots$; we refer to $\varepsilon_1, \varepsilon_2, \ldots$ as the elementary events in the σ-field $\mathcal{F}$.*

Remark 8.2. *An intuitive view of Definition 8.4 can be obtained by associating with each $\mathcal{F}_i$ a partition of Ω into blocks $B_1^i, B_2^i, \ldots$ such that the events B_j^i generate the σ-field $\mathcal{F}_i$. Furthermore, the partition associated with $\mathcal{F}_{i+1}$ is a refinement of partition associated with $\mathcal{F}_i$, and $\mathcal{F}_0$ is generated by the trivial partition while $\mathcal{F}_n$ is generated by the partition of Ω into the singleton sets containing the sample points.*

Definition 8.6. (Alon and Spencer [1992]) *A martingale is a sequence $X_0, \ldots, X_n$ of random variables so that for $0 \leq i < n$,*

$$\mathbf{E}[X_{i+1} | X_i, X_{i-1}, \ldots, X_0] = X_i.$$

Finally, for the sake of completeness, we adopt the following definitions from Chung and Lu [2006] in order to introduce Definition 8.9, which is equivalent to Definition 8.6 for the finite cases.

Definition 8.7. *If $f : \Omega \to \mathbb{R}$ is a function, we define the expectation $\mathbf{E}[f] = \mathbf{E}[f(x) | x \in \Omega]$ by*

$$\mathbf{E}[f] = \mathbf{E}[f(x) | x \in \Omega] := \sum_{x \in \Omega} f(x) \mathbf{Pr}[x].$$

Definition 8.8. *If $\mathcal{F}$ is a σ-field on Ω, we define the conditional expectation* $\mathbf{E}[f|\mathcal{F}] := \Omega \to \mathbb{R}$ *by*

$$\mathbf{E}[f|\mathcal{F}](x) = \frac{1}{\sum_{y\in\mathcal{F}(x)}\mathbf{Pr}[y]}\sum_{y\in\mathcal{F}(x)} f(y)\mathbf{Pr}[y],$$

where $\mathcal{F}(x)$ is the smallest element of $\mathcal{F}$ that contains x.

Definition 8.9. *A martingale obtained from random variable X is associated with a filter $\mathbf{F}$: $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \ldots \subseteq \mathcal{F}_n = 2^{\Omega}$ and a sequence of random variables $X_0, X_1, \ldots, X_m$ satisfying*

$$X_i = \mathbf{E}[X|\mathcal{F}_i], \tag{8.2.7}$$

and in particular, $X_0 = \mathbf{E}[X]$ and $X_m = X$.

8.3 Proof Techniques and Some Notation

We first introduce some notation regarding the simple probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$ as follows. The set Ω is the set of all possible outcomes for $2NK$ pairs of random bits, where we denote each pair with $b(j,k)$ for individual j at position k. The σ-field $\mathcal{F}$ of events is the set $\Sigma(\Omega)$ of all subsets of Ω; and the probability measure $\mathbf{Pr}$ is based on the product of probabilities of each pair of random bits $b(j,k), \forall j,k$ corresponding to Bernoulli(p_a^k), where $a \in \{1,2\}$ depends on the population of origin for individual j. Formally,

Definition 8.10. *The elementary events in the underlying sample space $(\Omega, \mathcal{F}, \mathbf{Pr})$ are all possible 4^{2NK} choices of $n = 2NK$ pairs of bits. For $0 \le i \le n$ and $w \in \{00,01,10,11\}^i$, let B_w denote the event that the first i pairs of bits equal to the bit string w. Let $\mathcal{F}_i$ be the σ-field generated by the partition of Ω into blocks B_w, for $w \in \{00,01,10,11\}^i$. Then the sequence $\mathcal{F}_0, \ldots, \mathcal{F}_n$ forms a filter. In the σ-field $\mathcal{F}_i$, the only valid events are the ones that depend on the values of the first i pairs, and all such events are valid within.*

The events that we define next and their interactions are shown in Figure 8.3.2.

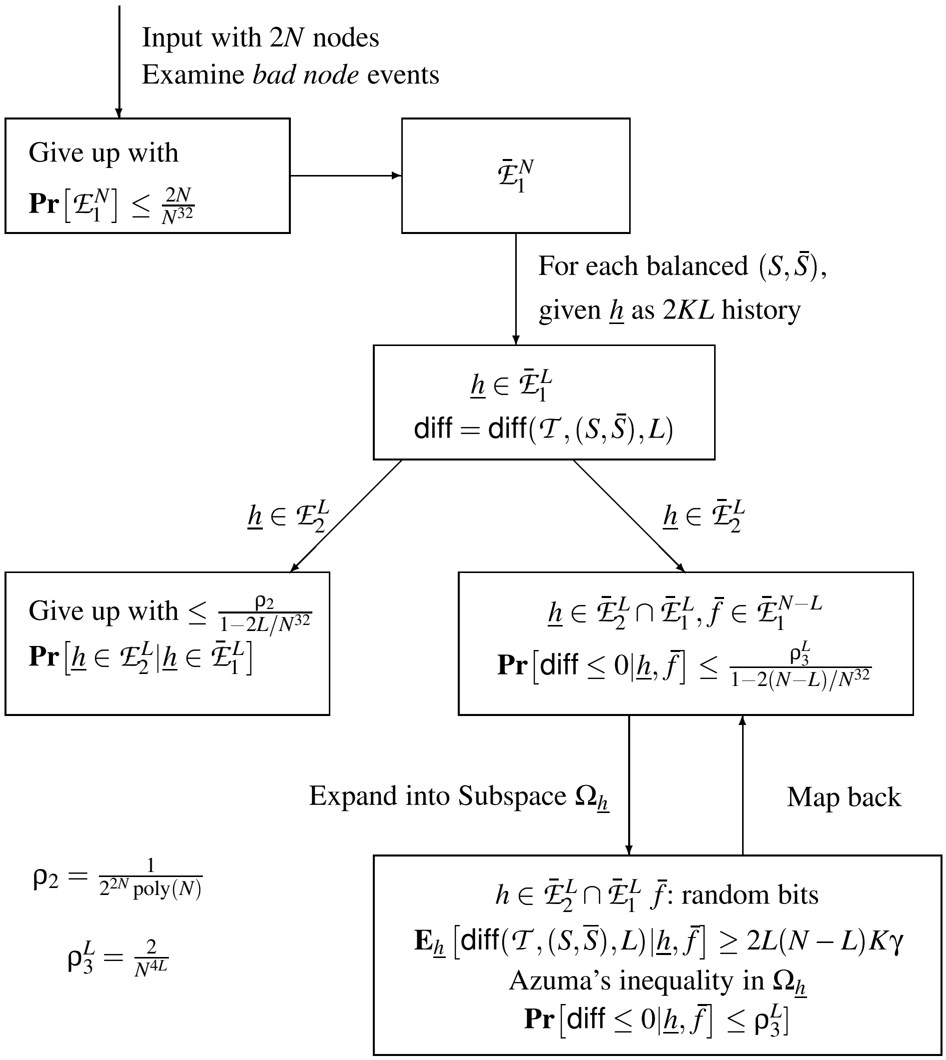

Figure 8.3.2. EVENTS RELATIONSHIP IN CHAPTER 8

We show that, with high probability, all of the $O(2^{2N})$ random variables $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)$, as in (8.1.5), one corresponding to each balanced $(S,\bar{S})$, are positive.

What we do is the following: we initially confine ourselves into a *good* subspace $\bar{\mathcal{E}}_1^N$ by excluding any *bad node* event. We then use union bound to bound the possibility of any *bad score* event, where a single *bad score* event occurs when $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0$ for a particular balanced cut $(S,\bar{S})$.

Each time we examine $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)$ for a particular balanced cut $(S,\bar{S})$, we let vector $(H_1,\ldots,H_{2KN})$ record the entire history of random unordered pairs of bits, where $(H_1,\ldots,H_{2KL})$ record the partial history of unordered pairs for the $2L$ swapped nodes corresponding to $(S,\bar{S})$. Let $\ell = 2KL$ be a positive integer. We denote this $2KL$-history with $\underline{H}^{(\ell)}$. Let $\underline{h}$ be a fixed possible ℓ-history.

We use the bounded differences method to bound a single *bad score* event over $(S,\bar{S})$: $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0$. Our starting point is after we reveal the $2KL$ unordered pairs and obtain a $2KL$-history $\underline{h}$. For simplicity of analysis, we first *expand* the confined subspace $\bar{\mathcal{E}}_1^N$ given $\underline{h}$, by dropping constraints on the $2(N-L)$ unswapped nodes. In this expanded subspace, we only require that the first $2L$ swapped nodes are *good* nodes, a condition that we denote with $\bar{\mathcal{E}}_1^L(S,\bar{S})$, while leaving the remaining $2(N-L)$ unswapped nodes to take completely random bits according to their distributions; that is, these nodes can be *bad* nodes. We then obtain a bound on concentration for $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)$ in this expanded probability space given $\underline{h}$. Eventually we map the probability of the bad score event that corresponds to random variable $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)$ from this expanded probability space back to the original confined subspace $\bar{\mathcal{E}}_1^N$ given $\underline{h}$.

For now, let us first introduce some notation for convenience and see how we expand the subspace $\bar{\mathcal{E}}_1^N$ given a particular history $\underline{h}$.

We call the remaining $2K(N-L)$ unordered pairs as the $2K(N-L)$-*future*. Let $\bar{f} = (H_{2KL+1}, \ldots, H_{2KN})$ be a fixed possible $2K(N-L)$-*future*.

Let $\Omega_{\underline{h}}$ denote that event that we observe this particular $2KL$-history: $\Omega_{\underline{h}} = \{\pi \in \Omega : H^{(\ell)}(\pi) = \underline{h}\}$. Given that event $\Omega_{\underline{h}}$ occurs, we are concerned about the following probability space $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$, where $\mathbf{Pr}_{\underline{h}}$ is the probability measure on $\Omega_{\underline{h}}$. Let us use $\mathbf{E}_{\underline{h}}$ for expectation in this space. Formally,

Definition 8.11. $\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T}, (S, \overline{S}), L)\right] = \mathbf{E}\left[\mathit{diff}(\mathcal{T}, (S, \overline{S}), L) \,|\, \mathcal{F}_{2KL}\right]$ *is the expected value of* $\mathit{diff}(\mathcal{T}, (S, \overline{S}), L)$ *conditioned on an event* $\underline{h} \in \mathcal{F}_{2KL}$. *This conditional expectation* $\mathbf{E}\left[\mathit{diff}(\mathcal{T}, (S, \overline{S}), L) \,|\, \mathcal{F}_{2KL}\right]$ *is a random variable that can be viewed as a function into reals from the blocks in the partition of* $\mathcal{F}_{2KL}$. *Hence* $\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T}, (S, \overline{S}), L)\right]$ *is an evaluation of this conditional expectation at a particular outcome* $\underline{h} \in \mathcal{F}_{2KL}$.

Thus $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$ corresponds to the *expanded* subspace of $\bar{\mathcal{E}}_1^N$ given $\underline{h}$; in this *expanded* probability space, we can apply the bounded differences method to analyze probability for a bad score event on $\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L)$ for a balanced cut $(S, \bar{S})$ in a clean manner.

In fact, our starting point of the bounded differences analysis is $\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L)\right]$, where $\underline{h}$ is a fixed possible $2KL$-history that we record while revealing all $2KL$ random unordered pairs on the $2L$ swapped nodes for $(S, \bar{S})$, subject to $\underline{h} \in \bar{\mathcal{E}}_1^L(S, \bar{S})$. This immediately indicates that the conditional expected value $\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L)\right] \geq 2(N-L)LK\gamma$, which is our "advantageous base point" given that $\Omega_{\underline{h}}$ occurs. Now as we reveal one by one the future $2K(N-L)$ random unordered pairs, the conditional expected values $\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L) | \underline{H}^{(\ell')}\right], \forall \ell' \geq 2KL$ form a martingale that is amenable to the bounded differences analysis.

This naturally brings up the second bad event $\mathcal{E}_2^L$ that we need to further exclude from the $2KL$-history $\underline{h}$, while examining a balanced cut $(S, \bar{S})$ in probability space $\bar{\mathcal{E}}_1^N$. $\mathcal{E}_2^L$ refers to the event that large deviation occurs simultaneously across a set of K random variables, where the k^{th} random variable is defined over the $2L$ unordered pairs of bits at locus k across the $2L$ swapped nodes.

Note that despite the additional constraint we put over $\underline{h} \in \bar{\mathcal{E}}_1^L \cap \bar{\mathcal{E}}_2^L$, $\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L)\right]$ remains lower bounded by $2(N-L)LK\gamma$, given that $\underline{h} \in \bar{\mathcal{E}}_1^L$ and $\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L)\right]$ is a random variable whose outcome is entirely determined by $\underline{h}$ (see Proposition 8.2).

However, excluding $\mathcal{E}_2^L$ from $\underline{h}$ is crucial in bounding the difference that each of the $2(N-L)K$-*future* random pairs of bits causes when we work in probability space $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$, where the *difference* refers to $\left|\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{(\ell')}\right] - \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)|\underline{H}^{(\ell'-1)}\right]\right|$, where $2KN \geq \ell' > 2KL$ depends on the particular pair of bits, such that the square sum of all these differences is not too big. This allows us to bound the probability on a bad score event, i.e., $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0$, using Azuma's inequality in probability space $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$, given that $\Omega_{\underline{h}}$ occurs, where $\underline{h} \in \bar{\mathcal{E}}_1^L \cap \bar{\mathcal{E}}_2^L$.

After we obtain $\mathbf{Pr}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0\right]$ in probability space $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$, where $\underline{h} \in \bar{\mathcal{E}}_1^L \cap \bar{\mathcal{E}}_2^L$ and $\bar{f}$ is entirely at random, we can calculate $\mathbf{Pr}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0\right]$ given that $\underline{h} \in \bar{\mathcal{E}}_1^L \cap \bar{\mathcal{E}}_2^L$ and $\bar{f} \in \bar{\mathcal{E}}_1^{N-L}$, where $\bar{\mathcal{E}}_1^{N-L}$ denote the event that the $2(N-L)$ unswapped nodes contain no *bad* node event either; hence the latter conditions imply that all nodes are drawn from $\mathcal{E}_1^N$.

Since $\mathbf{Pr}_{\underline{h}}\left[\mathcal{E}_1^{N-L}\right]$ is small, its influence on $\mathbf{Pr}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0\right]$ is small; that is, given that $\Omega_{\underline{h}}$ occurs, where $\underline{h} \in \bar{\mathcal{E}}_1^L \cap \bar{\mathcal{E}}_2^L$, $\mathbf{Pr}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0\right]$ remains small regardless whether $\bar{f}$ stays in this confined future subspace $\bar{\mathcal{E}}_1^{N-L}$ or is entirely at random as in $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$.

Let us map these notation to what we have defined in Section 7.3 regarding the expected difference of two conditional independent random variables as follows, where expectations are taken over all $2K(N-L)$ random unordered pairs of bits on $X_i, Y_i, \forall i$ after fixing swapped nodes $U_j, V_j, \forall j \in [1,L]$ for the given balanced cut.

Proposition 8.2. *We work in probability space* $(\Omega, \mathcal{F}, \mathbf{Pr})$. $\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\overline{S}),L)\right]$ *is a function of* $\underline{h}$*; Hence* $\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\overline{S}),L)\right]$ *as a random variable according to Definition 8.11, its value depends on random unordered pairs on the* $2L$ *swapped nodes that we record in* $\underline{h} = \{\tilde{U}_1, \ldots, \tilde{U}_L, \tilde{V}_1, \ldots, \tilde{V}_L\}$*:*

$$\begin{aligned}
\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\overline{S}),L)\right] &= \sum_{j=1}^{L}\sum_{i=1}^{N-L} \mathit{diff}(U_j) + \sum_{j=1}^{L}\sum_{i=1}^{N-L} \mathit{diff}(V_j) \\
&= (N-L)\sum_{j=1}^{L}\sum_{k=1}^{K} 2(p_2^k - p_1^k)(f^k(U_j) - f^k(V_j)),
\end{aligned}$$

where $\mathit{diff}(U_j)$ *and* $\mathit{diff}(V_j)$ *are defined in Definition 7.6.*

Proof. For each balanced cut $(S,\bar{S})$, we could pair up $X_i, Y_i, \forall i \in [1, N-L]$ via an arbitrary matching between two sets of unswapped nodes. Note that actual choice of i for X_i, Y_i does not influence the value of random variable $\mathsf{diff}(U_j)$, which is uniquely determined by the K unordered pairs of bits on node U_j.

Given (8.1.5) and linearity of expectations, we have

$$\begin{aligned}
\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)\right] &= \mathbf{E}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L) | U_j = \tilde{U}_j, V_j = \tilde{V}_j, \forall j \in [1,L]\right] \\
&= \sum_{j=1}^{L}\sum_{i=1}^{N-L} \mathbf{E}_{Y_i}\left[\mathsf{Pscore}(\tilde{U}_j, Y_i)\right] - \mathbf{E}_{X_i}\left[\mathsf{Pscore}(\tilde{U}_j, X_i)\right] + \\
&\quad \sum_{j=1}^{L}\sum_{i=1}^{N-L} \mathbf{E}_{X_i}\left[\mathsf{Pscore}(\tilde{V}_j, X_i)\right] - \mathbf{E}_{Y_i}\left[\mathsf{Pscore}(\tilde{V}_j, Y_i)\right] \\
&= \sum_{j=1}^{L}\sum_{i=1}^{N-L} \mathsf{diff}(U_j) + \sum_{j=1}^{L}\sum_{i=1}^{N-L} \mathsf{diff}(V_j) \\
&= (N-L)\sum_{j=1}^{L}\sum_{k=1}^{K} 2(p_2^k - p_1^k)(f^k(U_j) - f^k(V_j)), \qquad (8.3.8)
\end{aligned}$$

where the last equation is due to (7.3.15) and (7.3.16), given that $\forall j$, $U_j \sim \vec{p}_1$ and $V_j \sim \vec{p}_2$,

$$\mathsf{diff}(U_j) = \sum_{k=1}^{K} 2(p_2^k - p_1^k) f^k(U_j) + S, \qquad (8.3.9)$$

$$\mathsf{diff}(V_j) = \sum_{k=1}^{K} 2(p_1^k - p_2^k) f^k(V_j) - S. \qquad (8.3.10)$$

$\square$

Remark 8.3. *Hence* $\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\overline{S}),L)\right]$*, as a function of* $\underline{h}$*, is evaluated to a unique value, i.e., the value of* $(N-L)\sum_{j=1}^{L}\sum_{k=1}^{K} 2(p_2^k - p_1^k)(f^k(U_j) - f^k(V_j))$ *evaluated at outcome* $\{\tilde{U}_1, \dots, \tilde{U}_L, \tilde{V}_1, \dots, \tilde{V}_L\}$.

8.4 Excluding Two Bad Events

There are two lemmas regarding two types of bad events that we exclude from the set of all possible $\underline{h}$ that we consider in order to apply the bounded differences analysis. In more detail, after we record the ℓ-history $\underline{h}$, by revealing $U_j, V_j, \forall j \in [1, L]$ to something like $\tilde{U}_j, \tilde{V}_j, \forall j$, we can observe that with high probability, the history $\underline{h}$ excludes two types of bad events, from all balanced $(S, \bar{S})$.

– The first bad event is $\mathcal{E}_1^N$, which is defined in Definition 8.13. We show that by excluding a set of $2N$ *bad node* events (Definition 8.12) from the product probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$ simultaneously, each regarding a single node's behavior across its K unordered pairs of bits, $\mathcal{E}_1^N$ does not happen.

  In Section 8.4.1, we show that for all balanced cut $(S, \bar{S})$, when evaluated at a particular history $\underline{h} \in F_{2KL}$ that is determined by the $2KL$ unordered pairs on the $2L$ swapped nodes drawn from $\bar{\mathcal{E}}_1^N$,

$$\mathbf{E}_{\underline{h}} \left[\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L) | \underline{h} \in \bar{\mathcal{E}}_1^L, \bar{f} \text{ at random} \right] \geq 2KL(N - L)\gamma, \qquad (8.4.11)$$

  in an *expanded* subspace where $\bar{f}$ is assumed to be at random, given $\underline{h} \in \bar{\mathcal{E}}_1^L$.

– The second type of bad events $\mathcal{E}_2^L$, as in Definition 8.15, is regarding *simultaneously large* deviation across a set of K random variables defined over $2L$ swapped nodes and across their K loci. The idea is that, at each locus, we may observe certain large deviation from the expected bit pattern across $2L$ individuals in the sense of Definition 8.14; however, as we show in Lemma 8.4, with high probability, such deviation across all K loci can not be *simultaneously* large due to the mutual independence assumption that we make across K loci. We use union bound to bound $\mathcal{E}_2^L$ over all balanced $(S, \bar{S})$.

  (8.4.11) still holds given that the random variable $\mathbf{E}_{\underline{h}} \left[\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L) \right]$ is a function of $\underline{h} \in \bar{\mathcal{E}}_1^L \cap \bar{\mathcal{E}}_2^L$, with $\bar{f}$ entirely at random.

We formally define these two events with the following definitions.

Definition 8.12. (Bad Node Event $\mathcal{E}(Z)$) *Let a* bad node *event $\mathcal{E}(Z)$ be the event that $\mathsf{diff}(Z) < K\gamma$, where Z is one sample point. Note this is an event in an individual probability space $(\Omega_Z, \mathcal{F}_Z, \mathbf{Pr}_Z)$, where $(\Omega_Z, \mathcal{F}_Z, \mathbf{Pr}_Z)$ is defined over all possible outcomes for K random unordered pairs of bits for an individual Z.*

Note that all bad node events are mutually independent.

From now on, we use $(\Omega_i, \mathcal{F}_i, \mathbf{Pr}_i)$ to refer to $(\Omega_{Z_i}, \mathcal{F}_{Z_i}, \mathbf{Pr}_{Z_i})$ for the input $2N$ nodes, assuming that we are given a unique ordered list of $(Z_1, \ldots, Z_{2N})$.

Definition 8.13. (Bad Event $\mathcal{E}_1^N$) *$\mathcal{E}_1^N$ is the same as $\mathcal{E}(Z_1) \cup \ldots \cup \mathcal{E}(Z_{2N})$ in the product probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$ composed of distinct probability spaces $(\Omega_1, \mathcal{F}_1, \mathbf{Pr}_1), \ldots, (\Omega_{2N}, \mathcal{F}_{2N}, \mathbf{Pr}_{2N})$ as in Definition 8.12. Hence $\bar{\mathcal{E}}_1^N$ is the same as the joint event $\bar{\mathcal{E}}(Z_1) \cap \ldots \cap \bar{\mathcal{E}}(Z_{2N})$ in $(\Omega, \mathcal{F}, \mathbf{Pr})$.*

Let $(S, \bar{S})$ denote a balanced cut with L swapped nodes on each side for some $L \in [1, N/2]$. We proceed to define $\mathcal{E}_2^L$ given a balanced cut $(S, \bar{S})$. We first define a set of K random variables and their deviation, each regarding $2L$ unordered pairs across the $2L$ swapped nodes at a particular locus $k, \forall k$. Again let $\underline{h}$ be the $2KL$-history that we record after revealing bits on $2L$ swapped nodes in $(S, \bar{S})$.

Definition 8.14. (Deviation Values) *$\forall k = 1, \ldots, K$, let $t_k\sqrt{L}$ be the* exact *deviation of the following random variable that we observe over $\underline{h}$, which we denote with $f_2^k(\underline{h})$, i.e., $f_2^k(\underline{h}) - \mathbf{E}\left[f_2^k(\underline{h})\right] = t_k\sqrt{L}, \forall k$, where*

$$\begin{aligned} f_2^k(\underline{h}) &= f_2^k(U_1, \ldots, U_L, V_1, \ldots, V_L) \\ &= \sum_{j=1}^{L} (I_{00}^k(U_j) - I_{11}^k(U_j)) - (I_{00}^k(V_j) - I_{11}^k(V_j)) \\ &= \sum_{j=1}^{L} f^k(U_j) - \sum_{j=1}^{L} f^k(V_j), \end{aligned}$$

and $f^k(U_1), \ldots, f^k(U_L), f^k(V_1), \ldots, f^k(V_L)$ are all random variables in range $[-1, 1]$, as defined in Definition 7.4.

First let us obtain the expected value of $f_2^k(\underline{h})$.

Proposition 8.3. *For $f_2^k(\underline{h})$ as in Definition 8.14,*

$$\begin{aligned} \mathbf{E}\left[f_2^k(\underline{h})\right] &= \mathbf{E}\left[\sum_{j=1}^{L} \left[(I_{00}^k(U_j) - I_{11}^k(U_j)) - (I_{00}^k(V_j) - I_{11}^k(V_j))\right]\right] \\ &= \sum_{j=1}^{L} \mathbf{E}\left[f^k(U_j)\right] - \sum_{j=1}^{L} \mathbf{E}\left[f^k(V_j)\right] \\ &= 2L(p_2^k - p_1^k). \end{aligned}$$

Next we bound the deviation for each random variable $f_2^k(\underline{h})$, as defined in Definition 8.14.

Lemma 8.1. *$\forall k$, for random variable $f_2^k(\underline{h})$, we have*

$$\mathbf{Pr}\left[\left|f_2^k(\underline{h}) - \mathbf{E}\left[f_2^k(\underline{h})\right]\right| \geq t_k\sqrt{L}\right] \leq 2e^{-t_k{}^2/4}. \tag{8.4.12}$$

In addition, events corresponding to different loci are independent.

Proof. Let us define random variables $\bar{U}^k, \bar{V}^k$ such that

$$f_2^k(\underline{h}) = L(\bar{U}^k - \bar{V}^k), \tag{8.4.13}$$

and

$$\bar{U}^k = \sum_{j=1}^{L} f^k(U_j)/L, \qquad \bar{V}^k = \sum_{j=1}^{L} f^k(V_j)/L.$$

Thus by Proposition 8.3,

$$\mathbf{E}\left[\bar{U}^k\right] - \mathbf{E}\left[\bar{V}^k\right] = \frac{1}{L}\mathbf{E}\left[f_2^k(\underline{h})\right] = 2(p_2^k - p_1^k). \tag{8.4.14}$$

In order to bound probability of deviation on both sides of the expected differences, we let $t = t_k\sqrt{L}/L$ and apply Corollary 7.1 of Theorem 7.1,

$$\begin{aligned}
\mathbf{Pr}\left[\left|f_2^k(\underline{h}) - \mathbf{E}\left[f_2^k(\underline{h})\right]\right| \geq t_k\sqrt{L}\right] = \\
\mathbf{Pr}\left[\left|\bar{U}^k - \bar{V}^k - (\mathbf{E}\left[\bar{U}^k\right] - \mathbf{E}\left[\bar{V}^k\right])\right| \geq t_k\sqrt{L}/L\right] && (8.4.15)\\
\leq \quad 2e^{\frac{-2(t_k\sqrt{L}/L)^2}{(2/L)(2)^2}} && (8.4.16)\\
\leq \quad 2e^{-t_k^2/4}. && (8.4.17)
\end{aligned}$$

□

Definition 8.15. (Bad Deviation Event $\mathcal{E}_2^L$) *In probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$, given a balanced cut $(S, \bar{S})$ and its corresponding $2KL$-history $\underline{h}$, $\mathcal{E}_2^L$ is the event such that the set of random variables $t_1, \ldots, t_k$ regarding $2KL$ unordered pairs recorded in $\underline{h}$,* *as defined in Definition 8.14, are* simultaneously large *and satisfy*

$$\sum_{k=1}^{K} t_k^2 \geq \Lambda = 32N\ln 2 + 16K\ln 2(\log\log N + 1) + 6\ln N.$$

Using Definition 8.15 and 8.14, we immediately have the following lemma, which we use in Section 8.5.

Lemma 8.2. *Given that $\underline{h} \in \bar{\mathcal{E}}_2^L$, we have $\forall k$,*

$$\left|f_2^k(\underline{h})\right| \leq \left|\mathbf{E}\left[f_2^k(\underline{h})\right]\right| + \left|t_k\sqrt{L}\right|,$$

and

$$\sum_{k=1}^{K} t_k^2 \leq \Lambda,$$

where t_k is defined in Definition 8.14, and the bad deviation event $\mathcal{E}_2^L$ is given in Definition 8.15.

Proof. By definition of $t_k, \forall k$, we have that $f_2^k(\underline{h}) = \mathbf{E}\left[f_2^k(\underline{h})\right] + t_k\sqrt{L}$, where $t_k \in [\frac{-2L-\mathbf{E}\left[f_2^k(\underline{h})\right]}{\sqrt{L}}, \frac{2L-\mathbf{E}\left[f_2^k(\underline{h})\right]}{\sqrt{L}}]$.

Thus we immediately have $\left|f_2^k(\underline{h})\right| \leq \left|\mathbf{E}\left[f_2^k(\underline{h})\right]\right| + \left|t_k\sqrt{L}\right|$, where $\sum_{k=1}^{K} t_k^2 \leq \Lambda$, given that $\underline{h} \in \bar{\mathcal{E}}_2^L$. □

We are now ready to bound the probability for events $\bar{\mathcal{E}}_1^N$ and $\bar{\mathcal{E}}_2^L$; We want to emphasize the we exclude $\bar{\mathcal{E}}_1^N$ once for all $2N$ nodes, while excluding one $\bar{\mathcal{E}}_2^L$ from each balanced cut $(S, \bar{S})$, where L denotes that the event $\bar{\mathcal{E}}_2^L$ is defined over the particular set of $2KL$ unordered pairs across K loci on the $2L$ swapped nodes in $(S, \bar{S})$; we have $\binom{N}{L}^2$ number of such events for each L, whose probabilities we sum up later using union bound.

Hence for all $\Omega_{\underline{h}}$ that we deal with later in bounded differences analysis, neither of the two types of bad events happen.

Lemma 8.3. *Let $K \geq \frac{256\ln N}{\gamma}$, in probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$, $\mathbf{Pr}\left[\mathcal{E}_1^N\right] \leq \rho_1 = \frac{2N}{N^{32}}$. Thus,*

$$\mathbf{Pr}\left[\bar{\mathcal{E}}_1^N\right] \geq (1 - \frac{1}{N^{32}})^{2N} \geq 1 - \frac{2N}{N^{32}}.$$

Proof. Apply Lemma 7.3 to each $\mathsf{diff}(Z)$ with $\tau = 1/N^{32}$; Given $K \geq \frac{256 \ln N}{\gamma}$, we have $\forall Z$,

$$\mathbf{Pr}_Z\left[\mathcal{E}(Z)\right] \leq \frac{1}{N^{32}}.$$

Given that at equilibrium, each node's bits at locus k are two independent random draws from its distribution Bernoulli(p_a^k), where $a \in \{1,2\}$ depends on its population of origin for node Z, we adopt the view of composing the product space $(\Omega, \mathcal{F}, \mathbf{Pr})$ through distinct probability spaces $(\Omega_1, \mathcal{F}_1, \mathbf{Pr}_1), \ldots, (\Omega_{2N}, \mathcal{F}_{2N}, \mathbf{Pr}_{2N})$ as in Definition 8.13, where $(\Omega_i, \mathcal{F}_i, \mathbf{Pr}_i), \forall i$, is defined over all possible outcomes for K random unordered pairs for individual Z_i.

Then for events $\bar{\mathcal{E}}(Z_1) \in \mathcal{F}_1, \ldots, \bar{\mathcal{E}}(Z_{2N}) \in \mathcal{F}_{2N}$, the probability of the joint event $(\bar{\mathcal{E}}(Z_1), \ldots, \bar{\mathcal{E}}(Z_{2N}))$,

$$\begin{aligned}&\mathbf{Pr}\left[\bar{\mathcal{E}}(Z_1) \cap \bar{\mathcal{E}}(Z_2) \cap \ldots \cap \bar{\mathcal{E}}(Z_{2N})\right] = \\ &\qquad \mathbf{Pr}_1\left[\bar{\mathcal{E}}(Z_1)\right] \cdot \mathbf{Pr}_2\left[\bar{\mathcal{E}}(Z_2)\right] \cdot \ldots \cdot \mathbf{Pr}_{2N}\left[\bar{\mathcal{E}}(Z_{2N})\right], \end{aligned} \tag{8.4.18}$$

where the product corresponds to performing independent experiments with respect to each of the $2N$ probability spaces $(\Omega_i, \mathcal{F}_i, \mathbf{Pr}_i), \forall i$, given a fixed ordering of $(Z_1, \ldots, Z_{2N})$ on the input nodes.

Therefore by definition,

$$\begin{aligned}
\mathbf{Pr}\left[\bar{\mathcal{E}}_1^N\right] &= \mathbf{Pr}[\text{none of } \mathcal{E}(Z) \text{ happens, for all nodes } Z] && (8.4.19)\\
&= \mathbf{Pr}\left[\bar{\mathcal{E}}(Z_1) \cap \bar{\mathcal{E}}(Z_2) \cap \ldots \cap \bar{\mathcal{E}}(Z_{2N})\right] && (8.4.20)\\
&= \mathbf{Pr}_1\left[\bar{\mathcal{E}}(Z_1)\right] \cdot \mathbf{Pr}_2\left[\bar{\mathcal{E}}(Z_2)\right] \cdot \ldots \cdot \mathbf{Pr}_{2N}\left[\bar{\mathcal{E}}(Z_{2N})\right] && (8.4.21)\\
&= (1 - \mathbf{Pr}_1[\mathcal{E}(Z_1)]) \cdot (1 - \mathbf{Pr}_2[\mathcal{E}(Z_2)]) \cdot \ldots \cdot (1 - \mathbf{Pr}_{2N}[\mathcal{E}(Z_{2N})]) &&\\
&\geq (1 - \frac{1}{N^{32}})^{2N} \geq 1 - \frac{2N}{N^{32}}. && (8.4.22)
\end{aligned}$$

□

Lemma 8.4. *In probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$, for each balanced cut $(S, \bar{S})$,*

$$\mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L\right] \leq \rho_2,$$

where $\rho_2 = O(\frac{1}{2^{2N}\mathrm{poly}(N)})$ and $N \geq 8$.

Proof. To facilitate our proof, we obtain a set of nonnegative numbers $(\tilde{t}_1,\ldots,\tilde{t}_k)$ as follows; $\forall k$, to obtain $\tilde{t}_k$, we round $|t_k|$ down to nearest nonnegative number $|\tilde{t_k}|$ that is power of two.

It is easy to verify that $\forall k, t_k \in \left[\frac{-2L-\mathbf{E}\left[f_2^k(\underline{h})\right]}{\sqrt{L}}, \frac{2L-\mathbf{E}\left[f_2^k(\underline{h})\right]}{\sqrt{L}}\right]$ by Proposition 8.3. Thus we have $\tilde{t}_k \le |t_k| \le \left|2\sqrt{L}\right| + \left|\frac{\mathbf{E}\left[f_2^k(\underline{h})\right]}{\sqrt{L}}\right|$.

Let us divide the entire range of $|t_k|$ into intervals using power-of-2 nonnegative integers as dividing points; Let $r_k, \forall k$ represent the number of such intervals: we have $\forall k$, so long as $N \ge 8$,

$$\begin{aligned} r_k &= \log(\left|2\sqrt{L}\right| + \left|2L(p_1^k - p_2^k)/\sqrt{L}\right|) && (8.4.23)\\ &\le \log 4\sqrt{L} \le \log 4\sqrt{N/2} \le \log N. && (8.4.24)\end{aligned}$$

Thus we have at most $(\log N)^K$ blocks in the K-dimensional space such that each block along each dimension is a subinterval of $\left[0, \left|2\sqrt{L}\right| + \left|\frac{\mathbf{E}\left[f_2^k(\underline{h})\right]}{\sqrt{L}}\right|\right]$.

Let $\mathbf{B}(\beta_1,\ldots,\beta_k)$ represent a block in the K-dimensional space, where $\beta_1,\ldots,\beta_k$ are nonnegative power-of-2 integers and every point in $\mathbf{B}(\beta_1,\ldots,\beta_k)$ has its value fixed in interval $[\beta_k, 2\beta_k)$ along dimension $k, \forall k$; hence $(\beta_1,\ldots,\beta_k)$ is the point in the K-dimensional space with the smallest coordinate in every dimension in $\mathbf{B}(\beta_1,\ldots,\beta_k)$.

A set of values $(t_1,\ldots,t_k)$ as in Definition 8.14 is mapped into one of these blocks uniquely as follows. We say a point $(t_1,\ldots,t_k)$ *maps to* $\mathbf{B}(\beta_1,\ldots,\beta_k)$, if $\forall k, 2\beta_k > |t_k| \ge \beta_k$, i.e., $(\tilde{t_1},\ldots,\tilde{t_k}) = (\beta_1,\ldots,\beta_k)$.

We first bound the following event using Lemma 8.5. Let us fix one block $\mathbf{B}(\beta_1,\ldots,\beta_k)$ for a fixed set of values $\beta_1,\ldots,\beta_k$ such that $\sum_{k=1}^K \beta_k^2 \ge \Lambda/4$.

Lemma 8.5. *Let* $\Lambda/4 = 8N\ln 2 + 4K(\ln 2)(\log\log N + 1) + (3\ln N)/2$ *as* Λ *is defined in Definition 8.15.*

$$\mathbf{Pr}\left[\underline{h} \textit{ maps to a particular } \mathbf{B}(\beta_1,\ldots,\beta_k) \textit{ s.t.} \sum_{k=1}^K \tilde{t}_k^2 \ge \Lambda/4\right] \le \frac{1}{2^{2N}\cdot(\log N)^K \cdot N^{3/2}}.$$

Proof. Let $t_1\sqrt{L},\ldots,t_k\sqrt{L}$ be the deviation that we observe in $\underline{h}$ for random variables $f_2^1(\underline{h}), f_2^2(\underline{h}),\ldots,f_2^k(\underline{h})$ as in Definition 8.14. If coordinates $(\tilde{t_1},\ldots,\tilde{t_k})$ of $\underline{h}$

maps to $(\beta_1,\ldots,\beta_k)$, we know that $\forall k, 2\beta_k \geq |t_k| \geq \beta_k$ given the definition of $\mathbf{B}(\beta_1,\ldots,\beta_k)$.

In addition, by Lemma 8.1, we know that

$$\mathbf{Pr}\left[\left|f_2^k(\underline{h}) - \mathbf{E}\left[f_2^k(\underline{h})\right]\right| \geq \beta_k\sqrt{L}\right] \leq 2e^{-\beta_k{}^2/4}, \tag{8.4.25}$$

and events corresponding to different loci are independent; Thus we have

$$\begin{aligned}
& \mathbf{Pr}\left[\underline{h} \text{ maps to a particular } \mathbf{B}(\beta_1,\ldots,\beta_k) \text{ s.t.} \sum_{k=1}^{K}\beta_k^2 \geq \Lambda/4\right] \\
= \; & \prod_{k=1}^{K}\mathbf{Pr}\left[2\beta_k\sqrt{L} \geq \left(\left|f_2^k(\underline{h}) - \mathbf{E}\left[f_2^k(\underline{h})\right]\right| = \left|t_k\sqrt{L}\right|\right) \geq \beta_k\sqrt{L} \text{ s.t.} \sum_{k=1}^{K}\beta_k^2 \geq \Lambda/4\right] \\
\leq \; & \prod_{k=1}^{K}\mathbf{Pr}\left[\left|f_2^k(\underline{h}) - \mathbf{E}\left[f_2^k(\underline{h})\right]\right| \geq \beta_k\sqrt{L} \text{ s.t.} \sum_{k=1}^{K}\beta_k^2 \geq \Lambda/4\right] \\
\leq \; & \prod_{k=1}^{K} 2e^{-\beta_k^2/4} \leq 2^K e^{-\frac{\sum_{k=1}^{k}\beta_k^2}{4}} \leq 2^K e^{-\Lambda/16} && (8.4.26) \\
\leq \; & 2^K e^{-(2N\ln 2 + K\ln 2(\log\log N + 1) + 3\ln N/2)} && (8.4.27) \\
= \; & \frac{2^K}{2^{2N}\cdot(2\log N)^K \cdot N^{3/2}} && (8.4.28) \\
= \; & \frac{1}{2^{2N}\cdot(\log N)^K \cdot N^{3/2}}. && (8.4.29)
\end{aligned}$$

□

Given that $t_k^2 \leq 4\tilde{t}_k^2, \forall k$, we know that $\sum_{k=1}^{K} t_k^2 \geq \Lambda$ implies that

$$\sum_{k=1}^{K}\tilde{t}_k^2 \geq \frac{1}{4}\sum_{k=1}^{K} t_k^2 \geq \Lambda/4.$$

Thus we have

$$\begin{aligned}
\mathbf{Pr}\left[\sum_{k=1}^{K} t_k^2 \geq \Lambda\right] &\leq \mathbf{Pr}\left[\sum_{k=1}^{K} \tilde{t}_k^2 \geq \Lambda/4\right] && (8.4.30)\\
&= \mathbf{Pr}\left[\underline{h} \text{ maps to some } \mathbf{B}(\beta_1,\ldots,\beta_k) \text{ s.t. } \sum_{k=1}^{K} \beta_k^2 \geq \Lambda/4\right].
\end{aligned}$$

This allows us to upper bound $\mathbf{Pr}\left[\mathcal{E}_2^L\right]$ with events regarding $\sum_{k=1}^{K} \tilde{t}_k^2$ as follows:

$$\begin{aligned}
\mathbf{Pr}\left[\mathcal{E}_2^L\right] &= \mathbf{Pr}\left[\bigcap_{k=1}^{K} (f_2^k(\underline{h}) - \mathbf{E}\left[f_2^k(\underline{h})\right] = t_k\sqrt{L}) \text{ s.t. } \sum_{k=1}^{K} t_k^2 \geq \Lambda\right] && (8.4.31)\\
&\leq \mathbf{Pr}\left[\underline{h} \text{ maps to some } \mathbf{B}(\beta_1,\ldots,\beta_k) \text{ s.t. } \sum_{k=1}^{K} \beta_k^2 \geq \Lambda/4\right]\\
&\leq \frac{(\log N)^K}{2^{2N}\cdot(\log N)^K\cdot N^{3/2}} && (8.4.32)\\
&\leq \frac{1}{2^{2N}\,\mathrm{poly}(N)}. && (8.4.33)
\end{aligned}$$

Hence the probability that the $2KL$ unordered pairs induce simultaneously large deviation for random variables $f_2^1(\underline{h}),\ldots,f_2^k(\underline{h})$, as in Definition 8.15, is at most $\rho_2 = O(\frac{1}{2^{2N}\mathrm{poly}(N)})$. □

This immediately implies the following corollary.

Corollary 8.1. *For all balanced* $(S,\bar{S})$ *with* $L, \forall 0 < L \leq N/2$*, swapped nodes on each side, with probability* $1 - 1/\mathrm{poly}(N)$ *in probability space* $(\Omega, \mathcal{F}, \mathbf{Pr})$*, the set of* $t_1,\ldots,t_k$ *as defined in Definition 8.14 over* $2KL$ *unordered pairs satisfy* $\sum_{k=1}^{K} t_k^2 \leq \Lambda$.

Hence, we have an advantageous bounded differences case as we enter the expanded probability space $\Omega_{\underline{h}}$ by excluding $\underline{h}$ that belong to $\mathcal{E}_1^N$ or $\mathcal{E}_2^L$ for all balanced $(S,\bar{S})$.

8.4.1 Preparation for Landing in Expanded Subspaces

We first give two definitions.

Definition 8.16. *$\mathcal{E}_1^L(S,\bar{S})$ is the same as $\mathcal{E}(U_1)\cup\ldots\cup\mathcal{E}(U_L)\cup\mathcal{E}(V_1)\cup\ldots\cup\mathcal{E}(V_L)$ in the product probability space composed of distinct probability spaces defined over nodes $U_1,\ldots,U_L,V_1,\ldots,V_L$ as in Definition 8.12.*

Definition 8.17. *$\mathcal{E}_1^{N-L}(S,\bar{S})$ is the same as $\mathcal{E}(X_1)\cup\ldots\cup\mathcal{E}(X_{N-L})\cup\mathcal{E}(Y_1)\cup\ldots\cup\mathcal{E}(Y_{N-L})$ in the product probability space composed of distinct probability spaces defined over nodes $X_1,\ldots,X_{N-L},Y_1,\ldots,Y_{N-L}$ as in Definition 8.12.*

Hence $\bar{\mathcal{E}}_1^L$ and $\bar{\mathcal{E}}_1^{N-L}$ imply that no *bad node* event happens in the appropriate product spaces thus defined. We omit $(S,\bar{S})$ from $\mathcal{E}_1^L(S,\bar{S})$ and $\mathcal{E}_1^{N-L}(S,\bar{S})$ when it is clear from the context.

Given a balanced cut $(S,\bar{S})$, $\underline{h}$ records a history on the $2KL$ unordered pairs on swapped nodes $U_1,\ldots,U_L,V_1,\ldots,V_L$.

Proposition 8.4. *Given all nodes are drawn from $\bar{\mathcal{E}}_1^N$, for any balanced cut $(S,\bar{S})$ and its particular $2KL$-history $\underline{h}$ that we record satisfy the following: $\underline{h}\in\bar{\mathcal{E}}_1^L(S,\bar{S})$.*

Proof. Given $\bar{\mathcal{E}}_1^N$, we know that the joint event $(\bar{\mathcal{E}}(Z_1),\ldots,\bar{\mathcal{E}}(Z_{2N}))$ must happen in the product probability space $(\Omega,\mathcal{F},\mathbf{Pr})$. Hence for all nodes $Z_1,\ldots,Z_{2N}$,

$$\mathsf{diff}(Z_i)\geq K\gamma, \tag{8.4.34}$$

simultaneously in the product probability space $(\Omega,\mathcal{F},\mathbf{Pr})$, where $\mathsf{diff}(Z_i)$ is a random variable solely determined by node Z_i's bits across K loci, before or after we ever reveal it.

In particular, for each balanced $(S,\bar{S})$, we focus on the product probability space that is composed of distinct probability spaces defined over swapped nodes $U_1,\ldots,U_L,V_1,\ldots,V_L$ as in Definition 8.16. After we reveal these $2KL$ bits on nodes $U_j,V_j,\forall j=1,\ldots,L$, by (8.4.34),

$$\begin{aligned}\mathsf{diff}(U_j) &\geq K\gamma,\forall j=1,\ldots,L, &(8.4.35)\\ \mathsf{diff}(V_j) &\geq K\gamma,\forall j=1,\ldots,L. &(8.4.36)\end{aligned}$$

due to (8.4.34). Thus we have $\underline{h}\in\bar{\mathcal{E}}_1^L(S,\bar{S})$. □

Definition 8.18. *We use $\bar{f}$ to denote the* future *of the $2(N-L)K$ random unordered pairs that we are going to reveal for the unswapped nodes on a given balanced cut*

$(S,\bar{S})$. Recall that once we are fixed to the probability space such that $\mathcal{E}_1^N$ does not happen, we know that both $\underline{h}$ and $\bar{f}$ are confined; the following two notation are equivalent:

$$
\begin{aligned}
(\underline{h} \in \bar{\mathcal{E}}_1^L(S,\bar{S})) \quad &\cap \quad (\bar{f} \in \bar{\mathcal{E}}_1^{N-L}(S,\bar{S})),\\
(\underline{h},\bar{f}) \quad &\in \quad \bar{\mathcal{E}}_1^N.
\end{aligned}
$$

Remark 8.4. *Another way of seeing $\bar{\mathcal{E}}_1^L(S,\bar{S})$ (with respect to a particular balanced cut $(S,\bar{S})$) is to view it as an event in the simple probability space $(\mathbf{\Omega},\mathcal{F},\mathbf{Pr})$, such that we put constraints only on the specific $2L$ swapped nodes defined on $(S,\bar{S})$ while leaving the $\bar{f}$ at random. Hence we have $\bar{\mathcal{E}}_1^N \subset \bar{\mathcal{E}}_1^L(S,\bar{S})$, in $(\mathbf{\Omega},\mathcal{F},\mathbf{Pr})$.*

Thus $\underline{h}$ as a $2KL$-history on nodes drawn from $\bar{\mathcal{E}}_1^N$ (the product probability space $(\mathbf{\Omega},\mathcal{F},\mathbf{Pr})$ excluding $\mathcal{E}_1^N$), must satisfy $\underline{h} \in \bar{\mathcal{E}}_1^L(S,\bar{S})$.

We leave this confined space given $\bar{\mathcal{E}}_1^N$ for now and explore the following *expanded* subspace, where we require $\underline{h} \in \bar{\mathcal{E}}_1^L$ while leaving the future $\bar{f}$ at random. $(\mathbf{\Omega}_{\underline{h}},\Sigma(\mathbf{\Omega}_{\underline{h}}),\mathbf{Pr}_{\underline{h}})$ corresponds to this *expanded* subspace, where $\underline{h} \in \bar{\mathcal{E}}_1^L$.

Lemma 8.6. *For a balanced cut $(S,\bar{S})$, given a particular $2KL$-history $\underline{h} \in F_{2KL}$ on the $2L$ swapped nodes such that $\underline{h} \in \bar{\mathcal{E}}_1^L$,*

$$\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\overline{S}),L)|\underline{h} \in \bar{\mathcal{E}}_1^L,\bar{f} \text{ at random}\right] \geq 2L(N-L)K\gamma, \tag{8.4.37}$$

where expectation is over all possible outcome of the $2(N-L)K$ random unordered pairs in $\bar{f}$ in probability space $(\mathbf{\Omega}_{\underline{h}},\Sigma(\mathbf{\Omega}_{\underline{h}}),\mathbf{Pr}_{\underline{h}})$.

Proof. For a balanced cut $(S,\bar{S})$, given $\underline{h} \in \bar{\mathcal{E}}_1^L$, where $\underline{h}$ records $2KL$ bits over swapped nodes $U_j,V_j,\forall j=1,\ldots,L$, by Definition 8.12,

$$
\begin{aligned}
\mathsf{diff}(U_j) \quad &\geq \quad K\gamma,\forall j=1,\ldots,L, &(8.4.38)\\
\mathsf{diff}(V_j) \quad &\geq \quad K\gamma,\forall j=1,\ldots,L. &(8.4.39)
\end{aligned}
$$

Thus, in subspace $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$, where $\bar{f}$ is *at random* and $\underline{h} \in \bar{\mathcal{E}}_1^L$, we have from Proposition 8.2,

$$\begin{aligned}
\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)\right] &= \sum_{j=1}^{L}\sum_{i=1}^{N-L}\mathsf{diff}(U_j) + \sum_{j=1}^{L}\sum_{i=1}^{N-L}\mathsf{diff}(V_j) \\
&= (N-L)\sum_{j=1}^{L}\mathsf{diff}(U_j) + (N-L)\sum_{j=1}^{L}\mathsf{diff}(V_j) \\
&\geq 2L(N-L)K\gamma.
\end{aligned}$$

□

Recall that $\bar{\mathcal{E}}_2^L$ is the event that no simultaneously large deviation happens across $2L$ individuals over their $2KL$ unordered pairs.

Corollary 8.2. *Given that $\underline{h} \in \bar{\mathcal{E}}_1^L \cap \bar{\mathcal{E}}_2^L$, and $\bar{f}$ is at random:*

$$\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\overline{S}),L)|\underline{h} \in \bar{\mathcal{E}}_1^L \cap \bar{\mathcal{E}}_2^L, \bar{f} \textit{ at random}\right] \geq 2L(N-L)K\gamma, \tag{8.4.40}$$

which holds so long as $\underline{h} \in \bar{\mathcal{E}}_1^L$.

We next bound $\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)\right]$ for all balanced $(S,\bar{S})$, where $\underline{h}$ is confined in $\bar{\mathcal{E}}_1^N$ and $\bar{\mathcal{E}}_2^L$, before we enter each individually *expanded* subspace $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$.

Theorem 8.2. *Assume that all nodes in our sample are drawn from $\bar{\mathcal{E}}_1^N$, the probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$ excluding $\mathcal{E}_1^N$, we have $\forall$ balanced cut $(S,\bar{S})$, where $\underline{h}$ is a particular $2KL$-history that corresponds to the $2L$ swapped nodes specified over $(S,\bar{S})$ with respect to $\mathcal{T}$,*

$$\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\overline{S}),L)\right] \geq 2L(N-L)K\gamma, \tag{8.4.41}$$

where the conditional expectation is over each of the individually expanded probability space $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$, given $\underline{h} \in \bar{\mathcal{E}}_1^L$.

This statement remains true after we require that $\underline{h} \in \bar{\mathcal{E}}_2^L$ in addition.

Proof. By Proposition 8.4, for each balanced cut $(S,\bar{S})$, we have

$$\underline{h} \in \bar{\mathcal{E}}_1^L(S,\bar{S}). \tag{8.4.42}$$

Now apply Corollary 8.2, given that $\underline{h} \in \bar{\mathcal{E}}_1^L(S,\bar{S}) \cap \bar{\mathcal{E}}_2^L$, we immediately have the theorem. □

Remark 8.5. *$\mathit{diff}(Z)$ is determined by node Z's bit pattern, which is the same when we observe it from every balanced cut, where it acts as a swapped node. Hence although we do have $O(2^N)$ balanced cuts, $\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\bar{S}),L)\right]$ for all balanced cuts are just determined by the $2N$ random variables $\mathit{diff}(Z_1),\ldots,\mathit{diff}(Z_{2N})$, each of which is determined by the genotype of an individual in our sample.*

Hence, during the entire analysis of $O(2^{2N})$ balanced cuts, we may reveal a node Z in many cuts, but every time we reveal it, it is the same node; and the random bits at each locus $k, \forall k$ are just random draws from their corresponding distribution (e.g.,Bernoulli(p_a^k) for an individual from population $a, \forall a = 1,2$), before we start to reveal them in any cut, or after we have revealed them many times.

After we exclude the *bad* event $\mathcal{E}_1^N$ from probability space $(\Omega,\mathcal{F},\mathbf{Pr})$, given any $2KL$-history $\underline{h}$ that corresponds to a balanced cut $(S,\bar{S})$, we have an advantageous "base point" as we enter an individually *expanded* probability space $(\Omega_{\underline{h}},\Sigma(\Omega_{\underline{h}}),\mathbf{Pr}_{\underline{h}})$, where $\underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L$. This is true for all balanced cuts.

8.5 The Bounded Differences Approach in an Expanded Subspace

In this section, we bound the deviation of random variable $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)$ for a particular balanced cut $(S,\bar{S})$; recall that we let vector $(H_1,\ldots,H_{2KN})$ record the entire history of random unordered pairs that we see, where $(H_1,\ldots,H_{2KL})$ record the $2KL$-history $\underline{H}^{(\ell)}$ on $2L$ swapped nodes.

First it is convenient to introduce some more notation: For $\ell' \geq 2KL$, we begin to reveal the random unordered pairs on unswapped nodes in $(S,\bar{S})$. The random variable $\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)|\underline{H}^{(\ell')}\right]$ depends on the random extension $\underline{H}^{(\ell')}$ of $\underline{h}$ observed. By definition $\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)|\underline{H}^{(\ell')}\right](\pi) = \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)|\underline{H}^{(\ell')} = \underline{h}'\right]$ for $\pi \in \Omega_{\underline{h}}$, where $\underline{h}' = \underline{H}^{(\ell')}(\pi)$; another notation for this is $\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)|\mathcal{F}\right]$ where $\mathcal{F}$ is the σ-field generated by $\underline{H}^{(\ell')}$ restricted to $\Omega_{\underline{h}}$. To prove the theorem, we introduce the following.

Lemma 8.7. (Azuma's Inequality) *Let $Z_0, Z_1, \ldots, Z_m = f$ be a martingale on some probability space, and suppose that $|Z_i - Z_{i-1}| \leq c_i$, $\forall i = 1, 2, \ldots, m$, then*

$$\mathbf{Pr}[|f - \mathbf{E}[f]| \geq t] \leq 2e^{-t^2/2\sigma^2},$$

where $\sigma^2 = \sum_{i=1}^{m} c_i^2$.

Theorem 8.3. *Let $\underline{h}$ be a possible $2KL$-history that we record for a balanced cut $(S,\bar{S})$ such that $\underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L$. Then, for $t > 0$, in probability space $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$ such that all future $2(N-L)K$ unordered pairs in $\bar{f}$ are completely at random,*

$$\mathbf{Pr}_{\underline{h}}\left[|\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{2KN}\right] - \mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\bar{S}),L)\right]| \geq t\right] \leq 2e^{-t^2/2\sigma^2},$$

where $\sigma^2 \leq 64L^2(N-L)K\gamma + 16L(N-L)\Lambda$, for all balanced $(S,\bar{S})$ with $0 < L \leq N/2$ swapped nodes.

Proof. We shall set up things to use Lemma 8.7.

We work in probability space $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$. We start to reveal the $2K(N-L)$ unordered pairs on unswapped nodes that are chosen independently at random, and rely on $2L$ swapped nodes having a good history $\underline{h}$, given that $\underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L$.

Given the σ-field $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}))$, with $\Sigma(\Omega_{\underline{h}}) = 2^{\Omega_{\underline{h}}}$, let us first define a filter $\mathbf{F}$.

Given the independent random unordered pairs $H_{2KL+1}, \ldots, H_{2KN}$. The filter is defined by letting $\mathcal{F}_i, \forall i = 1, \ldots, m$, where $m = 2K(N-L)$, be the σ-field generated by histories $\underline{H}^{(2KL+1)}, \ldots, \underline{H}^{(2KL+i)}$. We thus obtain a natural $\mathbf{F}$:

$$\{\emptyset, \Omega_{\underline{h}}\} = \mathcal{F}_0 \subset \mathcal{F}_1 \subset \ldots \subset \mathcal{F}_m = 2^{\Omega_{\underline{h}}},$$

where for $0 \leq i \leq m = 2K(N-L)$, $(\Omega_{\underline{h}}, \mathcal{F}_i)$ is a σ-field.

Hence $\mathbf{F}$ corresponds to the increasingly refined partitions of $\Omega_{\underline{h}}$ obtained from all the different possible extensions of the $2KL$-history $\underline{h}$.

We obtain a martingale for random variable $\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)$ such that: Let $Z_0 = \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)\right]$ and

$$\begin{aligned} Z_{\ell'-2KL} &= \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{(\ell')}\right] && (8.5.43)\\ &= \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)|\mathcal{F}_{\ell'-2KL}\right], && (8.5.44)\end{aligned}$$

where $\mathcal{F}_{\ell'-2KL}$ is the σ-field generated by $\underline{H}^{(\ell')}$ restricted to $\Omega_{\underline{h}}$ and $2KN \geq \ell' > 2KL$.

We let $H_{2KL+1},\ldots,H_{2KN}$ map to random unordered pairs on $X_i^1,\ldots,X_{N-L}^K,Y_i^1,\ldots Y_{N-L}^K$, where X_i^k or Y_i^k refers to an unordered pair of bits on locus k on individual X_i or Y_i respectively.

We first define the following, $\forall j = 1,2,\ldots,m$, where $m = 2K(N-L)$,

$$\left|Z_j - Z_{j-1}\right| = c_j. \qquad (8.5.45)$$

We also need to translate between c_j, where $j = 1,2,\ldots,m$, and $d_{i,k}(X_i)$ and $d_{i,k}(Y_i)$, $\forall i = 1,\ldots,N-L, k = 1,\ldots,K$ that correspond to unordered pairs on locus k of X_i and Y_i respectively. In particular, $\forall i, \forall k$, we let

$$\begin{aligned} c_{(i-1)K+k} &= d_{i,k}(X_i), && (8.5.46)\\ c_{(N-L+i-1)K+k} &= d_{i,k}(Y_i). && (8.5.47)\end{aligned}$$

Let $j = 2KL + (i-1)K + k - 1$, we have

$$\begin{aligned} &d_{i,k}(X_i) = \\ &\quad \left|\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{(j)},X_i^k\right] - \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{(j)}\right]\right|.\end{aligned}$$

And similarly, let $\ell' = 2KL + (N-L)K + (i-1)K + k - 1$, we have

$$\begin{aligned} &d_{i,k}(Y_i) = \\ &\quad \left|\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{(\ell')},Y_i^k\right] - \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{(\ell')}\right]\right|.\end{aligned}$$

We immediately have the following lemmas that we can plug into Azuma's inequality, where $d_{i,k}$ applies to both $d_{i,k}(X_i)$ and $d_{i,k}(Y_i)$.

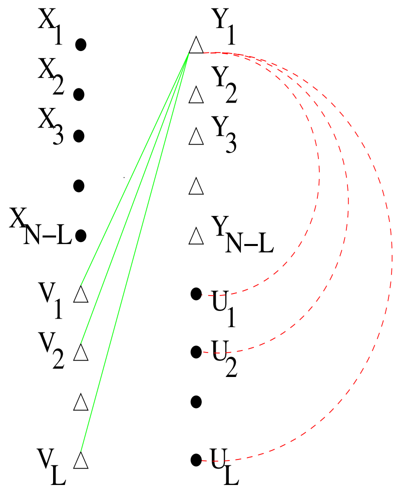

Figure 8.5.3. Set of edges that random unordered pairs on Y_1 influence upon

Lemma 8.8. *For the* $2(N-L)K$ *random unordered pairs on unswapped nodes* $X_i, Y_i\ \forall i \in [1, N-L]$ *that we reveal, at locus* $k \in [1, K]$*,*

$$d_{i,k} \leq \left|4L(p_2^k - p_1^k)\right| + \left|2t_k\sqrt{L}\right|,$$

where t_k *and* Λ *are defined in Definition 8.14 and Definition 8.15 respectively, and*

$$\sum_{k=1}^{K} t_k^2 \leq \Lambda.$$

Proof. Assume before we fix $Y_{i,k}$, we have reached history $\underline{H}^{(\ell')}$. W.l.o.g, assume the pair of random bits that we are fixing are on Y_i, and we let $Y_i = 00, 01/10, 11$ respectively and obtain the corresponding $d_{i,k}(Y_i), \forall i, k$; and we go through the same process to obtain $d_{i,k}(X_i), \forall i, k$.

Recall that by Definition 8.14, $f_2^k(\underline{h}) = \sum_{j=1}^{L}(I_{00}^k(U_j) - I_{11}^k(U_j)) - (I_{00}^k(V_j) - I_{11}^k(V_j))$, where $f_2^k(\underline{h})$ is defined in Definition 8.14 and $\left|\mathbf{E}\left[f_2^k(\underline{h})\right]\right| = \left|2L(p_2^k - p_1^k)\right|$ as shown in Proposition 8.3.

Thus by definition of $d_{i,k}(Y_i)$ and $d_{i,k}(X_i)$, we have

$$d_{i,k}(Y_i) = \begin{cases} \left|2p_2^k\right| \left|f_2^k(\underline{h})\right| & : \quad Y_i^k = 00 \\ \left|2q_2^k\right| \left|f_2^k(\underline{h})\right| & : \quad Y_i^k = 11 \\ \left|1-2p_2^k\right| \left|f_2^k(\underline{h})\right| & : \quad Y_i^k = 01/10, \end{cases}$$

and

$$d_{i,k}(X_i) = \begin{cases} \left|2p_1^k\right| \left|f_2^k(\underline{h})\right| & : \quad X_i^k = 00 \\ \left|2q_1^k\right| \left|f_2^k(\underline{h})\right| & : \quad X_i^k = 11 \\ \left|1-2p_1^k\right| \left|f_2^k(\underline{h})\right| & : \quad X_i^k = 01/10. \end{cases}$$

Note these changes reflect scores at locus k, which correspond to the set of edges in $\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)$, which are adjacent to Y_i and X_i respectively, as in Figure 8.5.3. Hence

$$d_{i,k}(Y_i) \quad \leq \quad 2\left|f_2^k(\underline{h})\right|, \tag{8.5.48}$$

$$d_{i,k}(X_i) \quad \leq \quad 2\left|f_2^k(\underline{h})\right|. \tag{8.5.49}$$

Thus given that $\underline{h} \in \bar{\mathcal{E}}_2^L$ and Lemma 8.2, we have

$$\begin{aligned} d_{i,k}(Y_i) \quad &\leq \quad 2\left|f_2^k(\underline{h})\right| & (8.5.50) \\ &\leq \quad 2\left(\left|\mathbf{E}\left[f_2^k(\underline{h})\right]\right| + \left|t_k\sqrt{L}\right|\right) & (8.5.51) \\ &= \quad \left|4L(p_2^k - p_1^k)\right| + \left|2t_k\sqrt{L}\right|, & (8.5.52) \end{aligned}$$

and similarly,

$$d_{i,k}(X_i) \quad \leq \quad \left|4L(p_2^k - p_1^k)\right| + \left|2t_k\sqrt{L}\right|, \tag{8.5.53}$$

where $\sum_{k=1}^{K} t_k^2 \leq \Lambda$. □

We are now ready to obtain a bound for $\sigma^2 = 2\sum_{i=1}^{N-L}\sum_{k=1}^{K} d_{i,k}^2$, where $d_{i,k}^2 \leq \left|4L(p_2^k - p_1^k)\right| + \left|2\sqrt{L}(t_k)\right|)^2$ applies to unswapped nodes $X_i, Y_i, \forall i = 1,\ldots,N-L, \forall k = 1,\ldots,K$ in bounding the differences they cause by revealing the unordered pairs on locus k.

Given that $\sum_{k=1}^{K} t_k^2 \leq \Lambda$,

$$
\begin{aligned}
\sigma^2 &= \sum_{i,k}(d_{i,k}^2(X_i) + d_{i,k}^2(Y_i)) = 2\sum_{i,k} d_{i,k}^2 \\
&\leq 2\sum_{i=1}^{N-L}\sum_{k=1}^{K}\left(\left|4L(p_2^k - p_1^k)\right| + \left|2\sqrt{L}(t_k)\right|\right)^2 \\
&\leq 2(N-L)\sum_k 2(4L(p_2^k - p_1^k))^2 + 2(2\sqrt{L}(t_k))^2 \\
&= 64L^2(N-L)\sum_k (p_2^k - p_1^k)^2 + 16L(N-L)\sum_k t_k^2 \\
&\leq 64(N-L)L^2(K\gamma) + 16(N-L)L\Lambda,
\end{aligned}
$$

where $\Lambda = 32N\ln 2 + 16K\ln 2(\log\log N + 1) + 6\ln N$ as in Definition 8.15. $\square$

We now apply Theorem 8.3 to obtain the following bound on a bad event. Note that the constant in the following lemma has not been optimized.

Lemma 8.9. *Let $\underline{h}$ be the specific $2KL$-history that we record for a balanced cut $(S,\bar{S})$ such that $\underline{h} \in \bar{\mathcal{E}}_1^L \cap \bar{\mathcal{E}}_2^L$. Let $\rho_3^L = \frac{2}{N^{4L}}$. Then,*

$$\mathbf{Pr}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0 | \underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L, \bar{f} \text{ at random}\right] \leq \rho_3^L,$$

given that $K \geq \Omega(\frac{\ln N}{\gamma})$ and $KN \geq \Omega(\frac{\ln N \log\log N}{\gamma^2})$, and $N \geq 4$.

Proof. By Theorem 8.3 with $t = \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)\right] \geq 2KL(N-L)\gamma$,

$$
\begin{aligned}
&\mathbf{Pr}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0 | \underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L\right] \\
= \;& \mathbf{Pr}_{\underline{h}}\left[\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L) | \underline{H}^{2KN}\right] - \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)\right] \leq -\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)\right]\right] \\
\leq \;& 2e^{-t^2/2\sigma^2} \leq 2e^{-(2KL(N-L)\gamma)^2/2\sigma^2}, \qquad (8.5.54)
\end{aligned}
$$

where σ^2 is defined in Theorem 8.3.

In the following calculation, we assume $N \geq 4$ at various places, but not any larger. Given that $\log\log N \geq 1, \forall N \geq 4$, we first rewrite σ^2 as the following:

$$
\begin{aligned}
\sigma^2 &\leq 64(N-L)L^2(K\gamma) + 16(N-L)L\Lambda \qquad (8.5.55) \\
&\leq 64K(N-L)L^2\gamma + 512\ln 2K(N-L)L\log\log N + 16(N-L)L(32N\ln 2 + 6\ln N).
\end{aligned}
$$

We will prove that for all N, so long as

(1) $K \geq \Omega(\frac{\ln N}{\gamma})$,

(2) $KN \geq \Omega(\frac{\ln N \log\log N}{\gamma^2})$,

we will have

$$2e^{-t^2/2\sigma^2} \leq 2e^{-(2KL(N-L)\gamma)^2/2\sigma^2} \leq \frac{2}{N^{4L}}. \tag{8.5.56}$$

In what follows, we show that given different values of N, by choosing slightly different constants in (1) and (2), (8.5.56) is always satisfied.

Case 1: $4 \leq N \leq \log\log N/2\gamma$**.**

In this case, we require that $KN \geq \frac{c_1 \ln N \log\log N}{\gamma^2}$, where $c_1 \geq 1488$, which immediately implies the following inequalities given that $N \leq \log\log N/2\gamma$:

(1) $K \geq \frac{2c_1 \ln N}{\gamma}$,

(2) $N \leq \frac{K \log\log N}{4c_1 \ln N}$,

(3) $\log\log N \geq 4\gamma, \forall N \geq 4$, i.e., we consider cases where γ is small enough,

(4) $\ln N \geq 2\ln 2$, $\forall N \geq 4$.

We first derive the following term that appears in σ^2 as in (8.5.55):

$$\begin{aligned}
16L(N-L)(32N\ln 2 + 6\ln N) &\leq 512\ln 2(N-L)LN + 96(N-L)L\ln N \\
&\leq \frac{128\ln 2K(N-L)L\log\log N}{c_1 \ln N} + \frac{48\gamma K(N-L)L}{c_1} \\
&\leq \frac{64K(N-L)L\log\log N}{c_1} + \frac{12K(N-L)L\log\log N}{c_1} \\
&\leq \frac{76K(N-L)L\log\log N}{c_1} \\
&\leq K(N-L)L\log\log N,
\end{aligned}$$

given that $c_1 \geq 1488$.

Next, given that $L\gamma \le N\gamma/2 \le \frac{\log\log N}{4}$, we have

$$\begin{aligned}\sigma^2 &\le 64K(N-L)L(L\gamma)+355K(N-L)L\log\log N+KL(N-L)\log\log N\\ &\le 16KL(N-L)\log\log N+356KL(N-L)\log\log N\\ &\le 372KL(N-L)\log\log N.\end{aligned}$$

Finally, given that $KN \ge \frac{1488\log\log N\ln N}{\gamma^2}$, we have:

$$\begin{aligned}2e^{-t^2/2\sigma^2} &\le e^{-(2KL(N-L)\gamma)^2/2\sigma^2}\\ &\le 2e^{-\frac{4KL(N-L)\gamma^2}{2\times 284\log\log N}} \le 2e^{-\frac{LKN\gamma^2}{284\log\log N}}\\ &\le \frac{2}{N^{4L}}.\end{aligned}$$

Thus we also have $K \ge \frac{2c_1\ln N}{\gamma} = \frac{2976\ln N}{\gamma}$ given that $N \le \log\log N/2\gamma$.

Case 2: $\frac{\log\log N}{2\gamma} < N \le \frac{K\log\log N}{20}$**.**

In this case, K and N are close and we require the following,

(1) $K \ge \frac{c_2\ln N}{\gamma}$, where $c_2 = 512$,

(2) $KN \ge \frac{c_0\ln N\log\log N}{\gamma^2}$, where $c_0 = 2000$.

Note that constants c_0, c_2 above are not optimized; given any N, an optimal combination of c_0, c_2 will result in the lowest possible K given that $K \ge \max\{\frac{c_0\ln N\log\log N}{N\gamma^2}, \frac{c_2\ln N}{\gamma}\}$.

Given that $N \le \frac{K\log\log N}{20}$, we have:

$$\begin{aligned}16L(N-L)(32N\ln 2+6\ln N) &\le \frac{400}{20}K(N-L)L\log\log N\\ &\le 20K(N-L)L\log\log N,\end{aligned}$$

and hence

$$\begin{aligned}\sigma^2 &\le 64K(N-L)L^2\gamma+355K(N-L)L\log\log N+20K(N-L)L\log\log N\\ &\le 64(N-L)L^2K\gamma+375KL(N-L)\log\log N.\end{aligned}$$

The following inequalities are due to (1) and (2) respectively,

$$\frac{(2KL(N-L)\gamma)^2}{2*64K(N-L)L^2\gamma} \geq 16L\ln N, \tag{8.5.57}$$

$$\frac{(2KL(N-L)\gamma)^2}{2*375KL(N-L)\log\log N} \geq \frac{16}{3}L\ln N, \tag{8.5.58}$$

and thus

$$2\sigma^2 \leq \frac{(2KL(N-L)\gamma)^2}{16L\ln N} + \frac{(2KL(N-L)\gamma)^2}{16L\ln N/3} \tag{8.5.59}$$

$$\leq \frac{(2KL(N-L)\gamma)^2}{4L\ln N/3}, \tag{8.5.60}$$

and $2e^{-t^2/2\sigma^2} \leq 2e^{\frac{-(2KL(N-L)\gamma)^2}{2\sigma^2}} \leq 2e^{-4L\ln N} \leq 2/N^{4L}$.

Case 3: $N \geq \frac{K\log\log N}{20} \geq 16$.

Here we require that $K = \frac{c_3\ln N}{\gamma}$ for some c_3 to be determined. Thus we have $KN \geq \frac{c_3^2\ln^2 N\log\log N}{80\gamma^2}$, which satisfies the constraint of the form $KN \geq \Omega(\frac{\ln N\log\log N}{\gamma^2})$ as in other cases.

Given that $N \geq 4$, we have that $\ln N \geq 2\ln 2$ and hence

$$\begin{aligned} 16L(N-L)(32N\ln 2 + 6\ln N) &\leq 128(N-L)LN\ln N + 6NL(N-L)\ln N \\ &\leq 134(N-L)LN\ln N. \end{aligned}$$

Given that $K\log\log N \leq 20N$, we have:

$$\begin{aligned} \sigma^2 &\leq 64K(N-L)L^2\gamma + 512\ln 2*(K\log\log N)(N-L)L + 134(N-L)LN\ln N \\ &\leq 64(N-L)L^2(K\gamma) + 512\ln 2*20N(N-L)L + 102(N-L)LN\ln N \\ &\leq 64\left(\frac{c_3\ln N}{\gamma}\right)\gamma(N-L)L(N/2) + (N-L)LN\ln N(128*20+134) \\ &\leq (32c_3+2694)(N-L)LN\ln N. \end{aligned}$$

By taking $c_3 = 188$ such that $c_3^2 \geq 4(32c_3 + 2694)$, we have

$$\begin{aligned}
t^2/2\sigma^2 &\geq \frac{(2K(N-L)L\gamma)^2}{2\sigma^2} = \frac{(2c_3(N-L)L\ln N)^2}{2\sigma^2} \\
&\geq \frac{2(c_3(N-L)L\ln N)^2}{(32c_3+2694)N(N-L)L\ln N} \\
&\geq \frac{2c_3^2(N-L)L\ln N}{(32c_3+2694)N} \\
&\geq \frac{c_3^2 L\ln N}{(32c_3+2694)} \geq 4L\ln N.
\end{aligned}$$

Thus $2e^{-t^2/2\sigma^2} \leq 2e^{-\frac{c_3^2 L\ln N}{(32c_3+2694)}} \leq 2e^{-4L\ln N} = \frac{2}{N^{4L}}$.

In summary, we have the following requirements. Note that N always falls into one of these cases. For all cases, we require that $K \geq \Omega(\ln N/\gamma)$ (which is implicit for Case 1); the constant that we require in K for Case 2 is larger than that for Case 3, (i.e., $c_2 \geq c_3$ as in above), so that the two cases can overlap.

- **Case 1:** $16 \leq N \leq \log\log N/2\gamma$. We require that $KN \geq \frac{1488\ln N\log\log N}{\gamma^2}$, which implies that $K \geq 2976\ln N/\gamma$.
- **Case 2:** $\frac{\log\log N}{2\gamma} < N \leq \frac{K\log\log N}{20}$. We require that $K \geq \frac{512\ln N}{\gamma}$, and $KN \geq \frac{2000\ln N\log\log N}{\gamma^2}$.
- **Case 3:** $N \geq \frac{K\log\log N}{20}$. We require $K \geq \frac{188\ln N}{\gamma}$.

□

In summary, instead of bounding the deviation of a random variable $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L)$ in a complete random space with all nodes taking random unordered pairs across all loci, we first reveal pairs of bits on all swapped nodes $U_j, V_j, \forall j \in [1,L]$ and record $\underline{h}$ for each $(S,\bar{S})$. We exclude two bad events $\mathcal{E}_1^L, \mathcal{E}_2^L$ from $\underline{h}$ for each $(S,\bar{S})$ in the original probability space $(\Omega,\mathcal{F},\mathbf{Pr})$, where all $2NK$ bits are at random. All histories that we consider for all balanced $(S,\bar{S})$ are good from this point on, and we can then apply bounded differences analysis in $(\Omega_{\underline{h}},\Sigma(\Omega_{\underline{h}}),\mathbf{Pr}_{\underline{h}})$ for all balanced cuts.

8.6 Mapping Back to Product Space Given $\bar{\mathcal{E}}_N^1$

We first prove one utility lemma.

Lemma 8.10. *For all balanced* $(S,\bar{S})$, $\mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L | \underline{h} \in \bar{\mathcal{E}}_1^L\right] \leq \frac{\rho_2}{1-2L/N^{32}}$, *hence*

$$\mathbf{Pr}\left[\underline{h} \in \bar{\mathcal{E}}_2^L | \underline{h} \in \bar{\mathcal{E}}_1^L\right] \geq 1 - \frac{\rho_2}{1-2L/N^{32}}.$$

Proof. Given the following equations:

$$\begin{aligned}
\mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L\right] &= \\
&\quad \mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L | \underline{h} \in \mathcal{E}_1^L\right] \cdot \mathbf{Pr}\left[\underline{h} \in \mathcal{E}_1^L\right] + \\
&\quad \mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L | \underline{h} \in \bar{\mathcal{E}}_1^L\right] \cdot \mathbf{Pr}\left[\underline{h} \in \bar{\mathcal{E}}_1^L\right], \qquad (8.6.61) \\
\mathbf{Pr}\left[\underline{h} \in \bar{\mathcal{E}}_1^L\right] &= (1 - \frac{1}{N^{32}})^{2L} \geq 1 - 2L/N^{32}, \qquad (8.6.62)
\end{aligned}$$

we have:

$$\begin{aligned}
\mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L | \underline{h} \in \bar{\mathcal{E}}_1^L\right] &= \\
&\quad \frac{\mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L\right] - \mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L | \underline{h} \in \mathcal{E}_1^L\right] \cdot \mathbf{Pr}\left[\underline{h} \in \mathcal{E}_1^L\right]}{\mathbf{Pr}\left[\underline{h} \in \bar{\mathcal{E}}_1^L\right]} \qquad (8.6.63) \\
&\leq \frac{\mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L\right]}{\mathbf{Pr}\left[\underline{h} \in \bar{\mathcal{E}}_1^L\right]} \qquad (8.6.64) \\
&\leq \frac{\rho_2}{1-2L/N^{32}}. \qquad (8.6.65)
\end{aligned}$$

□

Lemma 8.11. *For a balanced cut* $(S,\bar{S})$, *where its history* $\underline{h}$ *is conditioned on* $\bar{\mathcal{E}}_1^N$,

$$\mathbf{Pr}\left[\mathit{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0 | \bar{\mathcal{E}}_1^N\right] \leq \frac{\rho_2}{1-2L/N^{32}} + \frac{\rho_3^L}{1-2(N-L)/N^{32}}.$$

Proof. By assumption of independence between node events,

$$\begin{aligned}
\mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L | \bar{\mathcal{E}}_1^N\right] &= \mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L | \underline{h} \in \bar{\mathcal{E}}_1^L \cap \bar{f} \in \bar{\mathcal{E}}_1^{N-L}\right] && (8.6.66)\\
&= \mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L | \underline{h} \in \bar{\mathcal{E}}_1^L\right] && (8.6.67)\\
&\leq \frac{\rho_2}{1-2L/N^{32}}. && (8.6.68)
\end{aligned}$$

When $\underline{h} \in \mathcal{E}_2^L$, we give up trying to bound $\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0$; hence

$$\begin{aligned}
& \mathbf{Pr}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0 | \bar{\mathcal{E}}_1^N\right]\\
\leq\ & \mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L | \bar{\mathcal{E}}_1^N\right] +\\
& \mathbf{Pr}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0 | (\underline{h},\bar{f}) \in \bar{\mathcal{E}}_1^N \cap \underline{h} \in \bar{\mathcal{E}}_2^L\right] \cdot \mathbf{Pr}\left[\underline{h} \in \bar{\mathcal{E}}_2^L | \bar{\mathcal{E}}_1^N\right]\\
\leq\ & \frac{\rho_2}{1-2L/N^{32}} + \frac{\rho_3^L}{1-2(N-L)/N^{32}},
\end{aligned}$$

where the last inequalities are due to Lemma 8.10 and Lemma 8.12. □

Lemma 8.12. *For a balanced cut* $(S,\bar{S})$*,*

$$\mathbf{Pr}\left[\mathit{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0 | (\underline{h},\bar{f}) \in \bar{\mathcal{E}}_1^N \cap \underline{h} \in \bar{\mathcal{E}}_2^L\right] \leq \frac{\rho_3^L}{1-\frac{2(N-L)}{N^{32}}}.$$

Proof. We use e_0 to replace $\{\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0\}$ and bound the following: $\mathbf{Pr}\left[e_0 | (\underline{h} \in \bar{\mathcal{E}}_1^L \cap \bar{\mathcal{E}}_2^L) \cap \bar{f} \in \bar{\mathcal{E}}_1^{N-L}\right]$, which is the same as the term in the statement of the lemma,

$$\begin{aligned}
& \mathbf{Pr}\left[e_0 | \underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L, \bar{f} \text{ at random}\right] =\\
& \quad \mathbf{Pr}\left[e_0 | (\underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L) \cap \bar{f} \in \bar{\mathcal{E}}_1^{N-L}\right] \cdot \mathbf{Pr}\left[\bar{f} \in \bar{\mathcal{E}}_1^{N-L} | \underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L\right] +\\
& \quad \mathbf{Pr}\left[e_0 | (\underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L) \cap \bar{f} \in \mathcal{E}_1^{N-L}\right] \cdot \mathbf{Pr}\left[\bar{f} \in \mathcal{E}_1^{N-L} | \underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L\right].
\end{aligned}$$

By independence between node events:

$$\begin{aligned}
\mathbf{Pr}\left[\bar{f} \in \bar{\mathcal{E}}_1^{N-L} | \underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L\right] &= \mathbf{Pr}\left[\bar{f} \in \bar{\mathcal{E}}_1^{N-L}\right], && (8.6.69)\\
\mathbf{Pr}\left[\bar{f} \in \mathcal{E}_1^{N-L} | \underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L\right] &= \mathbf{Pr}\left[\bar{f} \in \mathcal{E}_1^{N-L}\right]. && (8.6.70)
\end{aligned}$$

Given that events $\mathcal{E}_2^L, \mathcal{E}_1^L$ defined on $2L$ swapped nodes are independent of event $\mathcal{E}_1^{N-L}$ on $2(N-L)$ unswapped nodes, we have the following, where we omit writing out the $\bar{f}$ at random condition,

$$\begin{aligned}
& \mathbf{Pr}\big[e_0 | (\underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L) \cap \bar{f} \in \bar{\mathcal{E}}_1^{N-L}\big] \\
= \; & \frac{\mathbf{Pr}\big[e_0 | \underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L\big] - \mathbf{Pr}\big[e_0 | (\underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L) \cap \bar{f} \in \mathcal{E}_1^{N-L}\big] \cdot \mathbf{Pr}\big[\bar{f} \in \mathcal{E}_1^{N-L}\big]}{\mathbf{Pr}\big[\bar{f} \in \bar{\mathcal{E}}_1^{N-L}\big]} \\
\leq \; & \frac{\mathbf{Pr}\big[\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L) \leq 0 | \underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L\big]}{\mathbf{Pr}\big[\bar{f} \in \bar{\mathcal{E}}_1^{N-L}\big]} \qquad (8.6.71) \\
\leq \; & \frac{\rho_3^L}{(1 - \frac{1}{N^{32}})^{2(N-L)}} \leq \frac{\rho_3^L}{(1 - \frac{2(N-L)}{N^{32}})}, \qquad (8.6.72)
\end{aligned}$$

where $\mathbf{Pr}\big[\bar{f} \in \bar{\mathcal{E}}_1^{N-L}\big] \geq 1 - \frac{2(N-L)}{N^{32}}$ following a proof similar to that of Lemma 8.3.

□

8.7 Putting Things Together

Finally, we prove Theorem 8.1.

Proof of theorem 8.1: Let $\mathcal{E}_3$ be the event that any balanced cut $(S, \bar{S})$ has a score higher than that of perfect partition $\mathcal{T}$,

$$\begin{aligned}
\mathbf{Pr}[\mathcal{E}_3] \; & \leq \; \mathbf{Pr}\big[E_1^N\big] + \sum_{(S,\bar{S})} \mathbf{Pr}\big[\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L) \leq 0 | \bar{\mathcal{E}}_1^N\big] \\
& \leq \; \left(1 - (1 - \frac{1}{N^{32}})^{2N}\right) + \sum_{(S,\bar{S})} \frac{\rho_2}{1 - 2L/N^{32}} + \frac{\rho_3^L}{1 - 2(N-L)/N^{32}} \\
& = \; \left(1 - (1 - \frac{1}{N^{32}})^{2N}\right) + \frac{2^{2N}\rho_2}{1 - 2L/N^{32}} + \\
& \quad \; \sum_{L=1}^{N/2} \binom{N}{L} \binom{N}{L} \frac{\rho_3^L}{1 - 2(N-L)/N^{32}} \\
& = \; O(1/\operatorname{poly}(N)),
\end{aligned}$$

where $\underline{h}$ is the specific $2KL$-history that we record for each $(S, \bar{S})$ after we exclude bad events $\mathcal{E}_1, \mathcal{E}_2$ from probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$. □

9 Learning Product Distributions

9.1 Introduction

After exploring the power of drawing two random vectors from its product distribution, i.e., two random draws from each of the K dimensional distributions, for each sample point, we ponder at the possibility of achieving the same power of clustering using a single random draw from each of the K dimensional distributions for each sample point. Let us first formally define a product distribution.

Definition 9.1. *A product distribution* $\mathbf{D}_m, \forall m = 1,2$*, over Boolean cube* $\{0,1\}^K$ *is characterized by its expected value* $\vec{p}_m = (p_m^1, \ldots, p_m^K) \in [0,1]^K$*, which we refer to as the center of* $\mathbf{D}_m$*.*

We then restate our problem as a fundamental problem of learning mixtures of two product distributions over discrete domains, in particular, over the K-dimensional Boolean cube $\{0,1\}^K$, where K is a variable whose value we need to resolve. Given a small sample, i.e., when N is small, can we learn the perfect partition with a small number of attributes from each sample point such that, for each attribute, we are given only a single bit that is randomly drawn from its corresponding Bernoulli distribution?

We finish this section by giving some more notation, followed by two results. We use $X = \vec{x} = (x^1, x^2, \ldots, x^K)$ to represent a random K-bit vector, given a set of K attributes. Sometimes we also use x_j^i to represent the i^{th} coordinate of point X_j.

Definition 9.2. *A random vector* $\vec{x}$ *from the distribution* $\vec{p}_m$*, which we denote as* $\vec{x} \sim \vec{p}_m$*, is generated by independently selecting each coordinate* x^i *to be* 1 *with probability* $p_m^i, \forall i, \forall m$*.*

We first define an alternative measure for the *average* distance between two product distributions $\mathbf{D}_1$, $\mathbf{D}_2$ across their K dimensions.

Definition 9.3. $\|\mathbf{D}_1 - \mathbf{D}_2\| = \frac{\|\vec{p}_1 - \vec{p}_2\|_1}{K} = \frac{\sum_{i=1}^{K} |p_1^i - p_2^i|}{K}$.

We prove in Section 9.2 Theorem 9.2, which holds under a special condition such that we know whether $p_1^i \geq p_2^i$, or vice versa, $\forall i$. Under this condition, Theorem 9.2 states a result that is similar to Theorem 7.2 (in Section 7.2), i.e., the Global Optimum Lemma, using only $K = \Omega(\ln N/\alpha^2)$ attributes, where $|\alpha| \geq \gamma$ is the same measure as in Definition 9.3. Similar to Theorem 7.2, we do not require a balanced input instance.

We next use the inner-product of two K-dimensional vectors $\vec{x}$ and $\vec{y}$ as the Rscore between X and Y, as in Definition 9.4, and define a complete graph where nodes are sample points and edge weight is the Rscore between the two points.

Definition 9.4. $\mathit{Rscore}(X,Y) = <\vec{x},\vec{y}> = \sum_{i=1}^{K} x^i y^i$.

Similar to Section 8.1, we define Rscore for a cut $(S,\bar{S})$ as the sum of Rscores over the set of edges in $(S,\bar{S})$. Let P_1 represent the set of points $X_1, X_2, \ldots, X_N$ from a product distribution $\mathbf{D}_1$, and P_2 represent the set of points $Y_1, Y_2, \ldots, Y_N$ from a product distribution $\mathbf{D}_2$. We show that the perfect partition $\mathcal{T} = (P_1, P_2)$ is the minimum cut (min-cut) in terms of Rscore among all balanced cut $(S,\bar{S})$, both in expectation and with high probability, despite the deviation of an individual Rscore or the sum of Rscores over a set of edges from its expectation. Formally,

Theorem 9.1. *Given $K = \Omega(\frac{\ln N}{\gamma})$ attributes from each of the $2N$ sample points, where N points come from each distribution, and $KN = \Omega(\frac{\ln N \log\log N}{\gamma^2})$, and $N \geq 4$, with probability $1 - 1/\mathrm{poly}(N)$, for all other balanced cut $(S,\bar{S})$ in the complete graph formed among $2N$ sample points,*

$$\mathit{Rscore}(\mathcal{T}) < \mathit{Rscore}(S,\bar{S}).$$

It is easy to check that, following the same line of arguments in this chapter for Theorem 9.1, using scores based on pairwise Hamming distances, i.e., $\forall X, Y$, $H(\vec{x},\vec{y}) = \sum_{i=1}^{K} x^i \oplus y^i$, the max-cut will identify the perfect partition with high probability, given the same order of number of attributes.

We also note that this inner-product based or Hamming distance based scores can not give results that are similar to the Global and Local Optimum lemmas in Chapter 7.

9.2 An Alternative Score

In this section, we prove Theorem 9.2, which is similar to Theorem 7.2, while requiring only a single bit from each attribute, under the condition that we know whether $p_1^i \geq p_2^i$, or vice versa, $\forall i$. We design a new score, which we call Bscore such that, with high probability, the absolute values of Bscores, each defined over $K = \Omega(\ln N/\alpha^2)$ attributes, between points from the same distribution are consistently lower than those between sample points from different distributions, where α is the same as the measure in Definition 9.3.

To simplify presentation, the Bscore we define in this section is based on the assumption that $p_1^i \geq p_2^i, \forall i$. When this assumption is not true, we only need to alter the definition of the Bscore such that for the set of attributes $p_1^i \geq p_2^i$, we add $+$ signs, while for the other set of attributes we add $-$ signs before Bscore^i that we currently define.

Definition 9.5. *For an unordered pair of points (X,Y), let*

$$\mathit{Bscore}^i(X,Y) = (x^i - y^i), \forall i, \tag{9.2.1}$$

and

$$\mathit{Bscore}(X,Y) = \sum_{i=1}^{K} \mathit{Bscore}^i(X,Y).$$

Under this assumption, the α as in Definition 9.6 is exactly the same measure as in Definition 9.3 and hence $|\alpha| \geq \gamma$.

Definition 9.6. $\alpha = (1/K)\sum_{i=1}^{K} \alpha^i$, *where* $\alpha^i = p_1^i - p_2^i$.

Definition 9.7. *For two sample points $X \in P_a$ and $Y \in P_b$, where $a, b \in \{1, 2\}$,*

$$\alpha(X,Y) = -\alpha(Y,X) = (1/K)\sum_{i=1}^{K} (p_x^i - p_y^i).$$

And hence $\alpha = |\alpha(X,Y)| = |\alpha(Y,X)|$.

Thus whether we subtract Y from X, or vice versa, it has a similar effect in terms of Theorem 9.2. We first show the following lemma.

Lemma 9.1. *Let X,Y come from different distributions, and Z_1,Z_2 be of common origin,*

$$\begin{aligned}
\mathbf{E}[\textit{Bscore}(X,Y)] &= K\alpha(X,Y),\\
\mathbf{E}[\textit{Bscore}(Z_1,Z_2)] &= 0.
\end{aligned}$$

Proof. Given $\alpha(X,Y)$ as in Definition 9.7,

$$\begin{aligned}
\mathbf{E}[\mathsf{Bscore}(X,Y)] &= \sum_{i=1}^{K}\mathbf{E}\left[\mathsf{Bscore}^i(X,Y)\right] && (9.2.2)\\
&= \sum_{i=1}^{K}\mathbf{E}\left[x^i - y^i\right] && (9.2.3)\\
&= \sum_{i=1}^{K}(p_x^i - p_y^i) && (9.2.4)\\
&= K\alpha(X,Y). && (9.2.5)
\end{aligned}$$

Given all frequencies are exactly the same for Z_1,Z_2, it follows that $\mathbf{E}[\mathsf{Bscore}(Z_1,Z_2)] = 0$. □

The following theorem does not assume a balanced input instance.

Theorem 9.2. *Let $2N$ be the size of the sample. Given that $K \geq 72\ln N/\alpha^2$, with probability $1 - O(1/N^2)$, for all points X,Y that come from different distributions, and for all points Z_1,Z_2 that come from the same distribution,*

$$\begin{aligned}
|\textit{Bscore}(X,Y)| &\geq 2K|\alpha|/3,\\
|\textit{Bscore}(Z_1,Z_2)| &\leq K|\alpha|/3.
\end{aligned}$$

Proof. We first use the Hoeffding bound to prove the following lemma.

W.l.o.g, assume that $X \in P_1$ and $Y \in P_2$ and hence $\alpha(X,Y) > 0$.

Lemma 9.2. *Given that $K \geq 72 \ln N/\alpha^2$,*

$$\begin{aligned}
\mathbf{Pr}[|\mathit{Bscore}(Z_1, Z_2)| \geq K\alpha/3] &< 2/N^4, \\
\mathbf{Pr}[|\mathit{Bscore}(X, Y)| \leq 2K\alpha/3] &< 2/N^4.
\end{aligned}$$

Proof. Given $\mathbf{E}[\mathsf{Bscore}(Z_1, Z_2)] = 0$ and $\mathsf{Bscore}(Z_1, Z_2) = \sum_{i=1}^{K} \mathsf{Bscore}^i(Z_1, Z_2)$ is the sum of K independent random variables with values in $[-1, 1]$, using Hoeffding bound as in Theorem 7.1, by taking $t = K\alpha/3K = \alpha/3$,

$$\mathbf{Pr}[|\mathsf{Bscore}(Z_1, Z_2)| \geq K\alpha/3] \leq 2e^{-2K^2(\alpha/3)^2/K(2)^2} \leq 2/N^4, \tag{9.2.6}$$

and similarly, given that $\mathbf{E}[\mathsf{Bscore}(Y, X)] = -\mathbf{E}[\mathsf{Bscore}(X, Y)] = -\alpha$, we use one of the following depending on the order of X, Y that we use for subtraction,

$$\begin{aligned}
&\mathbf{Pr}[\mathsf{Bscore}(X, Y) \leq 2K\alpha/3] = \\
&\qquad \mathbf{Pr}[-\mathsf{Bscore}(X, Y) + \mathbf{E}[\mathsf{Bscore}(X, Y)] \geq K\alpha/3] && (9.2.7) \\
&= \quad e^{-2K^2(\alpha/3)^2/K(2)^2} && (9.2.8) \\
&\leq \quad 1/N^4, && (9.2.9) \\
&\mathbf{Pr}[\mathsf{Bscore}(Y, X) \geq -2K\alpha/3] = \\
&\qquad \mathbf{Pr}[\mathsf{Bscore}(Y, X) - \mathbf{E}[\mathsf{Bscore}(Y, X)] \geq K\alpha/3] && (9.2.10) \\
&= \quad e^{-2K^2(\alpha/3)^2/K(2)^2} && (9.2.11) \\
&\leq \quad 1/N^4. && (9.2.12)
\end{aligned}$$

□

By union bound, for $\alpha > 0$, the probability that any event of type $|\mathsf{Bscore}(X, Y)| \leq 2K\alpha/3$ or type $|\mathsf{Bscore}(Z_1, Z_2)| \geq K\alpha/3$ happens is at most $8N^2/N^4$, since the total number of such events is $2N(2N - 1)$. Thus the theorem holds. □

Thus a simple threshold based algorithm will identify a perfect partition by taking that set of edges that have absolute values larger than a certain threshold in the complete graph. In particular, the perfect partition in such a graph has a maximum average score, i.e., the total score across edges in the perfect partition divided by the number of such edges.

Remark 9.1. *By definition of γ and α, it is easy to verify that $\gamma \geq \alpha^2$. Hence a bound of $K = \Omega(\ln N/\gamma)$ is tighter than that of $K = \Omega(\ln N/\alpha^2)$. We give bounds based on γ in Section 7.3 and Chapter 8, and also in the rest of this chapter.*

9.3 Overview of Key Ideas

We give an overview for the key ideas that we use to prove Theorem 9.1.

The Model and Notation. We use $\vec{x}$ to denote a K-bit vector that corresponds to X in the sample, and x^i its i^{th} component. We use $\vec{x} \sim \mathbf{D}_m, \forall m = 1,2$, to represent that x is a vector that is generated from the product distribution $\mathbf{D}_m$, and thus

$$\mathbf{E}_{\vec{x}\sim\mathbf{D}_m}[\vec{x}] = \vec{p}_m, \forall m = 1,2. \tag{9.3.13}$$

Recall that we build a complete graph such that sample points map to nodes in the graph and an edge between two nodes X and Y is given a weight of $\mathsf{Rscore}(X,Y)$ as in Definition 9.4.

The key idea that makes an inner-product based score work is that although from an individual sample point, e.g., Y's perspective, $\mathsf{diff}(Y)$ (which is similar to that in Definition 7.6, except that Pscores there are replaced with Rscores, see Definition 9.8,) may not be significant due to the definition of our Rscore, the sum of diffs over a pair of swapped nodes, e.g., $\mathsf{diff}(X)+\mathsf{diff}(Y)$ as in Figure 9.3.1, is significant despite a bounded amount of deviation. Indeed, we prevent the sum of $\mathsf{diff}(X)+\mathsf{diff}(Y)$ from deviating too much from its expected value, $K\gamma$, by excluding bad node events as in Definition 9.10, which is in essence similar to what we define in Definition 8.12, from X and Y.

Therefore, the advantage of a perfect partition over all other balanced cuts, i.e., the sum of diffs across $2L(N-L)$ pairs of edges from swapped nodes to unswapped nodes, as shown in Figure 8.1.1, is significant enough, so that the perfect partition can almost always win over all other balanced partitions, in terms of the particular measure (minimum total score here), despite the deviation events that we handle in Section 9.4.2.

The inspiration for using an inner-product based score and pairing up $\mathsf{diff}(X), \mathsf{diff}(Y)$ such that $X \sim \mathbf{D}_1$ and $Y \sim \mathbf{D}_2$ comes from Freund and Mansour [1999], where they did similar analysis up till Proposition 9.3. The rest of the proof follows exactly that in Chapter 8. And hence we only rewrite various propositions, lemmas and claims that have changed slightly due to the definition of this new score.

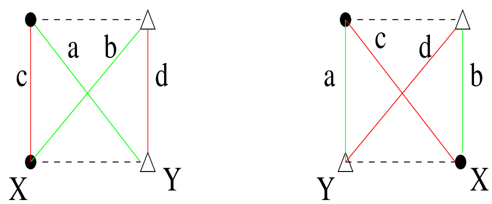

Figure 9.3.1. Given Dots$\sim \vec{p}_1$ and Triangles$\sim \vec{p}_2$. Define $\mathsf{diff}(X) = \mathbf{E}[c|X] - \mathbf{E}[b|X]$ and $\mathsf{diff}(Y) = \mathbf{E}[d|Y] - \mathbf{E}[a|Y]$. Given $K = \Omega(\ln N/\gamma)$, with high probability, $\mathsf{diff}(X) + \mathsf{diff}(Y) \geq K\gamma/2$, given that $\mathbf{E}_{\vec{x}\sim\vec{p}_1}[\mathsf{diff}(X)] + \mathbf{E}_{\vec{y}\sim\vec{p}_2}[\mathsf{diff}(Y)] = K\gamma$; Hence $a+b \leq c+d$, with high probability, given also that $KN = \Omega(\ln N \log\log N/\gamma^2)$.

9.3.1 The Expected Difference of Two Edges

Proposition 9.1. $\forall a, b = 1, 2, \mathbf{E}_{\vec{x}\sim\mathbf{D}_a, \vec{y}\sim\mathbf{D}_b}[<\vec{x}, \vec{y}>] = <\vec{p}_a, \vec{p}_b>$.

Proof. We have $\forall a, b = 1, 2$,

$$\begin{aligned} \mathbf{E}_{\vec{x}\sim\mathbf{D}_a, \vec{y}\sim\mathbf{D}_b}[<\vec{x}, \vec{y}>] &= \mathbf{E}\left[\sum_{i=1}^{K} x^i y^i\right] \\ &= \sum_{i=1}^{K} \mathbf{E}\left[x^i y^i\right] = \sum_{i=1}^{K} p_a^i p_b^i = <\vec{p}_a, \vec{p}_b> . \end{aligned}$$

□

Definition 9.8. *Let X be a sample point from distribution $\mathbf{D}_1$ and Y be a sample point from $\mathbf{D}_2$. Let X', Y' be points randomly drawn from $\mathbf{D}_1$ and $\mathbf{D}_2$ respectively,*

$$\begin{aligned} \mathit{diff}(X) &= \mathbf{E}_{\vec{x'}\sim\vec{p}_1}\left[\mathit{Rscore}(X, X')\right] - \mathbf{E}_{\vec{y'}\sim\vec{p}_2}\left[\mathit{Rscore}(X, Y')\right], \\ \mathit{diff}(Y) &= \mathbf{E}_{\vec{y'}\sim\vec{p}_2}\left[\mathit{Rscore}(Y, Y')\right] - \mathbf{E}_{\vec{x'}\sim\vec{p}_1}\left[\mathit{Rscore}(Y, X')\right], \end{aligned}$$

where expectations are taken over all possible realizations of X', Y' respectively.

Proposition 9.2. *Let X be a sample point from $\mathbf{D}_1$ and Y be a point from $\mathbf{D}_2$,*

$$\mathit{diff}(X) = \sum_{i=1}^{K} x^i (p_1^i - p_2^i), \qquad \mathit{diff}(Y) = \sum_{i=1}^{K} y^i (p_2^i - p_1^i).$$

Proof. By Definition 9.8, we have

$$\begin{aligned}
\mathsf{diff}(X) &= \mathbf{E}_{\vec{x}'\sim\vec{p}_1}\left[\mathsf{Rscore}(X,X')\right] - \mathbf{E}_{\vec{y}'\sim\vec{p}_2}\left[\mathsf{Rscore}(X,Y')\right] \\
&= \mathbf{E}_{\vec{x}'\sim\vec{p}_1}\left[< \vec{x},\vec{x}' >\right] - \mathbf{E}_{\vec{y}'\sim\vec{p}_2}\left[< \vec{x},\vec{y}' >\right] && (9.3.14) \\
&= < \vec{x},\vec{p}_1 - \vec{p}_2 > && (9.3.15) \\
&= \sum_{i=1}^{K} x^i (p_1^i - p_2^i), && (9.3.16) \\
\mathsf{diff}(Y) &= \mathbf{E}_{\vec{y}'\sim\vec{p}_2}\left[\mathsf{Rscore}(Y,Y')\right] - \mathbf{E}_{\vec{x}'\sim\vec{p}_1}\left[\mathsf{Rscore}(Y,X')\right] \\
&= \mathbf{E}_{\vec{y}'\sim\vec{p}_2}\left[< \vec{y},\vec{y}' >\right] - \mathbf{E}_{\vec{x}'\sim\vec{p}_1}\left[< \vec{y},\vec{x}' >\right] && (9.3.17) \\
&= < \vec{y},\vec{p}_2 - \vec{p}_1 > && (9.3.18) \\
&= \sum_{i=1}^{K} y^i (p_2^i - p_1^i). && (9.3.19)
\end{aligned}$$

□

We next show that the sum of two expected differences over X from $\mathbf{D}_1$ and Y from $\mathbf{D}_2$ is significant.

Proposition 9.3. $\mathbf{E}_{\vec{x}\sim\vec{p}_1}[\mathit{diff}(X)] + \mathbf{E}_{\vec{y}\sim\vec{p}_2}[\mathit{diff}(Y)] = \|\vec{p}_1 - \vec{p}_2\|_2^2 = K\gamma.$

Proof. By Proposition 9.2,

$$\begin{aligned}
\mathbf{E}_{\vec{x}\sim\vec{p}_1}[\mathsf{diff}(X)] + \mathbf{E}_{\vec{y}\sim\vec{p}_2}[\mathsf{diff}(Y)] &= \sum_{i=1}^{K} p_1^i (p_1^i - p_2^i) + \sum_{i=1}^{K} p_2^i (p_2^i - p_1^i) \\
&= < \vec{p}_1, \vec{p}_1 - \vec{p}_2 > + < \vec{p}_2, \vec{p}_2 - \vec{p}_1 > \\
&= K\gamma.
\end{aligned}$$

□

Hence let us denote w.l.o.g. $\eta = \mathbf{E}_{\vec{x}\sim\vec{p}_1}[\mathsf{diff}(X)] \geq K\gamma/2$, and thus $\mathbf{E}_{\vec{y}\sim\vec{p}_2}[\mathsf{diff}(X)] = K\gamma - \eta$.

We next show the following two claims.

Lemma 9.3. *Given that* $K \geq \frac{8\ln 1/\tau}{\gamma}$, $\mathbf{Pr}_X[\mathit{diff}(X) \geq \eta - K\gamma/4] > 1-\tau$.

Lemma 9.4. $\mathbf{Pr}_Y[\mathit{diff}(Y) \geq (K\gamma-\eta) - K\gamma/4] > 1-\tau$.

Proof of Lemma 9.3: Let us define $\gamma_k = (p_1^k - p_2^k)^2, \forall k = 1,\ldots,K$. Given that $x^1,\ldots,x^K$ are independent Bernoulli random variables and $(p_1^k - p_2^k)x^k$ is either in $[0,\sqrt{\gamma_k}]$ or $[-\sqrt{\gamma_k},0]$, $\forall k = 1,\ldots,K$, we apply Hoeffding bound as in Theorem 7.1 with $t = K\gamma/4K = \gamma/4$:

$$\begin{aligned}
\mathbf{Pr}_X\left[-\sum_{k=1}^{K}(p_1^k - p_2^k)x^k + \eta \geq K\gamma/4\right] &= \\
\mathbf{Pr}_X\left[\sum_{k=1}^{K}(p_1^k - p_2^k)x^k - \eta \leq -K\gamma/4\right]& \\
\leq\ & e^{-2K^2(\gamma/4)^2/\sum_{k=1}^{K}(\sqrt{\gamma_k})^2} \\
\leq\ & \tau.
\end{aligned}$$

Thus we have that $\mathbf{Pr}_X\left[\sum_{k=1}^{K}(p_1^k - p_2^k)x^k \geq \eta - K\gamma/4\right] \geq 1-\tau$. $\square$

Proof of Lemma 9.4: Similarly to proof of Lemma 9.3, we have

$$\mathbf{Pr}_Y\left[\sum_{k=1}^{K}(p_2^k - p_1^k)y^i - (K\gamma-\eta) \leq -K\gamma/4\right] \leq \tau,$$

where $K\gamma - \eta = \mathbf{E}_{\vec{y}\sim\vec{p}_2}[\mathsf{diff}(Y)]$. And hence

$$\mathbf{Pr}_Y\left[\sum_{k=1}^{K}(p_2^k - p_1^k)y^i \geq (K\gamma-\eta) - K\gamma/4\right] \geq 1-\tau.$$

$\square$

The rest of the proof for Theorem 9.1 follows the proof for Theorem 8.1 as presented in Chapter 8. Hence we only include changes and important steps.

9.4 Min-Cut Reveals the Perfect Partition

9.4.1 Probability Space

We first introduce some notation regarding the simple probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$ as follows. The set Ω is the set of all possible outcomes for $2NK$ random bits, where we denote each bit as b_j^k for a point j at dimension k.

Definition 9.9. *The elementary events in the underlying sample space $(\Omega, \mathcal{F}, \mathbf{Pr})$ are all possible 2^{2NK} choices of $n = 2NK$ bits. For $0 \leq i \leq n$ and $w \in \{0,1\}^i$, let B_w denote the event that the first i bits equal to the bit string w. Let $\mathcal{F}_i$ be the σ-field generated by the partition of Ω into blocks B_w, for $w \in \{0,1\}^i$. Then the sequence $\mathcal{F}_0, \dots, \mathcal{F}_n$ forms a filter. In the σ-field $\mathcal{F}_i$, the only valid events are the ones that depend on the values of the first i bits, and all such events are valid within.*

Remark 9.2. *Comparing the above definition and Definition 8.10, the only difference lies in whether a pair of bits or a single bit defines the random variable along a single dimension. For all other places: wherever the phrase "a pair of bits" or "pairs of bits" is used, it should be replaced with "a single bit" or "bits".*

We first show that the perfect partition has the minimum expected value among all balanced cuts, when summing up scores over all edges across the cut. We first modify the definition for a random variable $\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L)$ as in (8.1.5) to be:

$$\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L) = \mathsf{Rscore}(S, \bar{S}) - \mathsf{Rscore}(\mathcal{T}). \tag{9.4.20}$$

Proposition 9.4. $\mathbf{E}\left[\mathit{diff}(\mathcal{T}, (S, \overline{S}), L)\right] = (N - L)LK\gamma$.

Proof.

$$\begin{aligned} \mathbf{E}\left[\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L)\right] &= (N-L)L\mathbf{E}_{\vec{x}\sim\vec{p}_1}[\mathsf{diff}(X)] + \\ &\quad (N-L)L\mathbf{E}_{\vec{y}\sim\vec{p}_2}[\mathsf{diff}(Y)] \qquad (9.4.21) \\ &= (N-L)LK\gamma. \qquad (9.4.22) \end{aligned}$$

□

We work in probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$. For the rest of this section, for a balanced cut $(S, \bar{S})$, the $2KL$-history $\underline{h} = \{\tilde{U_1}, \ldots, \tilde{U_L}, \tilde{V_1}, \ldots, \tilde{V_L}\}$ is a random variable whose value depends on $2KL$ random bits on $2L$ swapped nodes specified over $(S, \bar{S})$ with respect to $\mathcal{T}$, as shown in Figure 8.1.1; recall that $\tilde{X}$ is the outcome of a particular point X in our sample.

We next restate Proposition 8.2 as the following.

Proposition 9.5. *As a random variable according to Definition 8.11,*

$$\begin{aligned}\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L)\right] &= (N-L)\sum_{j=1}^{L}\mathsf{diff}(U_j) + (N-L)\sum_{j=1}^{L}\mathsf{diff}(V_j)\\ &= (N-L)\sum_{j=1}^{L}\sum_{k=1}^{K}(p_1^k - p_2^k)(u_j^k - v_j^k),\end{aligned}$$

where $\mathsf{diff}(U_j)$ *and* $\mathsf{diff}(V_j)$ *are defined in Definition 9.8.*

We keep Definition 8.13 for Bad Event $\mathcal{E}_1^N$, except that we replace a Bad Node Event $E(Z), \forall Z$, with the following.

Definition 9.10. **(Bad Node Event)** *Let a* bad node *event* $\mathcal{E}(Z)$ *be the event that* $\mathsf{diff}(Z) < \mathbf{E}[\mathsf{diff}(Z)] - K\gamma/4$*, where Z is a sample point. Note this is an event in an individual probability space* $(\Omega_Z, \mathcal{F}_Z, \mathbf{Pr}_Z)$*, where* $(\Omega_Z, \mathcal{F}_Z, \mathbf{Pr}_Z)$ *is defined over all possible outcomes of K random bits for sample point Z.*

This immediately implies the following lemma, which only slightly modifies Lemma 8.6.

Lemma 9.5. *For a balanced cut* $(S, \bar{S})$*, given a particular* $2KL$*-history* $\underline{h} \in F_{2KL}$ *on the* $2L$ *swapped nodes such that* $\underline{h} \in \bar{\mathcal{E}}_1^L$*,*

$$\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T}, (S, \overline{S}), L) | \underline{h} \in \bar{\mathcal{E}}_1^L, \bar{f} \text{ at random}\right] \geq L(N-L)K\gamma/2, \tag{9.4.23}$$

where expectation is over all possible outcomes of the $2(N-L)K$ *random bits in* $\bar{f}$ *in probability space* $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$.

Proof. For a balanced cut $(S,\bar{S})$, given $\underline{h} \in \bar{\mathcal{E}}_1^L$, where $\underline{h}$ records $2KL$ bits over swapped nodes $U_j, V_j, \forall j = 1,\ldots,L$, by Definition 9.10,

$$\begin{aligned} \mathsf{diff}(U_j) &\geq \eta - K\gamma/4, \forall j = 1,\ldots,L, && (9.4.24) \\ \mathsf{diff}(V_j) &\geq K\gamma - \eta - K\gamma/4, \forall j = 1,\ldots,L, && (9.4.25) \end{aligned}$$

and hence $\mathsf{diff}(U_j) + \mathsf{diff}(V_j) \geq K\gamma/2, \forall j = 1,\ldots,L$. Thus, in $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$, where $\bar{f}$ is *at random* and $\underline{h} \in \bar{\mathcal{E}}_1^L$, we have from Proposition 9.5,

$$\begin{aligned} \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)\right] &= (N-L)\sum_{j=1}^{L}\mathsf{diff}(U_j) + (N-L)\sum_{j=1}^{L}\mathsf{diff}(V_j) \\ &\geq (N-L)\sum_{j=1}^{L}(\mathsf{diff}(U_j)+\mathsf{diff}(V_j)) \\ &\geq (N-L)LK\gamma/2. \end{aligned}$$

□

And thus we have the following theorem.

Theorem 9.3. *Give that all points are drawn from $\bar{\mathcal{E}}_1^N$, the probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$ excluding $\mathcal{E}_1^N$, we have $\forall$ balanced cut $(S,\bar{S})$, where $\underline{h}$ is a particular $2KL$-history corresponding to the $2L$ swapped nodes specified over $(S,\bar{S})$ with respect to $\mathcal{T}$,*

$$\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\overline{S}),L)\right] \geq (N-L)LK\gamma/2, \qquad (9.4.26)$$

where the conditional expectation is over each of the individually expanded probability space $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$ given $\underline{h} \in \bar{\mathcal{E}}_1^L$, where $\mathcal{E}_1^L$ is defined in Definition 8.16. This statement remains true after we require that $\underline{h} \in \bar{\mathcal{E}}_2^L$ in addition, where $\mathcal{E}_2^L$ is defined in Definition 9.13.

9.4.2 Large Deviation

We aim to define a Bad Deviation Event as in Definition 9.13, but we first overwrite some definitions regarding E_2^L.

Definition 9.11. *Given vectors $\vec{u}_1,\dots,\vec{u}_L$ and $\vec{v}_1,\dots,\vec{v}_L$, where u_j^k, v_j^k are the k^{th} bit of U_j and V_j respectively,*

$$f_2^k(\underline{h}) = f_2^k(U_1,\dots,U_L,V_1,\dots,V_L) = \sum_{j=1}^{L} u_j^k - \sum_{j=1}^{L} v_j^k.$$

Definition 9.12. (Deviation Values) *$\forall k = 1,\dots,K$, let $t_k\sqrt{L}$ be the* exact *deviation on $f_2^k(\underline{h})$, i.e., $f_2^k(\underline{h}) - \mathbf{E}\left[f_2^k(\underline{h})\right] = t_k\sqrt{L}, \forall k$.*

Definition 9.13. (Bad Deviation Event $\mathcal{E}_2^L$) *In probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$, given a balanced cut $(S,\bar{S})$ and its corresponding $2KL$-history $\underline{h}$, $\mathcal{E}_2^L$ is the event such that the set of random variables $t_1,\dots,t_k$ regarding $2KL$ random bits recorded in $\underline{h}$, as defined in Definition 9.12, are* simultaneously large *and satisfy*

$$\sum_{k=1}^{K} t_k^2 \geq \Delta = 8N\ln 2 + 4K\ln 2(\log\log N + 1) + 3\ln N/2.$$

Using Definition 9.13 and 9.12, we immediately have the following lemma.

Lemma 9.6. *Given that $\underline{h} \in \bar{\mathcal{E}}_2^L$, we have $\forall k$,*

$$\left|f_2^k(\underline{h})\right| \leq \left|\mathbf{E}\left[f_2^k(\underline{h})\right]\right| + \left|t_k\sqrt{L}\right|,$$

and $\sum_{k=1}^{K} t_k^2 \leq \Delta$, where t_k is in Definition 9.12, and $\mathcal{E}_2^L$ is in Definition 9.13.

Proof. By definition of $t_k, \forall k$, we have that $f_2^k(\underline{h}) = \mathbf{E}\left[f_2^k(\underline{h})\right] + t_k\sqrt{L}$, where $t_k \in \left[\frac{-L-\mathbf{E}\left[f_2^k(\underline{h})\right]}{\sqrt{L}}, \frac{L-\mathbf{E}\left[f_2^k(\underline{h})\right]}{\sqrt{L}}\right]$.

Thus we immediately have

$$\left|f_2^k(\underline{h})\right| \leq \left|\mathbf{E}\left[f_2^k(\underline{h})\right]\right| + \left|t_k\sqrt{L}\right|,$$

where $\sum_{k=1}^{K} t_k^2 \leq \Delta$, given that $\underline{h} \in \bar{\mathcal{E}}_2^L$. □

First let us obtain the expected value of $f_2^k(\underline{h}), \forall k$ as in Definition 9.11.

Proposition 9.6. $\mathbf{E}\left[f_2^k(\underline{h})\right] = \mathbf{E}\left[\sum_{j=1}^{L} u_j^k - v_j^k\right] = L(p_1^k - p_2^k)$.

Next we examine the deviation for each random variable $f_2^k(\underline{h}), \forall k$.

Lemma 9.7. *$\forall k$, for random variable $f_2^k(\underline{h})$ as in Definition 9.11,*

$$\mathbf{Pr}\left[\left|f_2^k(\underline{h}) - \mathbf{E}\left[f_2^k(\underline{h})\right]\right| \geq t_k\sqrt{L}\right] \leq 2e^{-t_k{}^2}. \tag{9.4.27}$$

In addition, events corresponding to different dimensions are independent.

Proof. Let us define random variables $\bar{U}^k$, $\bar{V}^k$ such that

$$f_2^k(\underline{h}) = L(\bar{U}^k - \bar{V}^k), \tag{9.4.28}$$

where $\bar{U}^k = \sum_{j=1}^L u_j^k/L$ and $\bar{V}^k = \sum_{j=1}^L v_j^k/L$.

Thus by Proposition 9.6,

$$\mathbf{E}\left[\bar{U}^k\right] - \mathbf{E}\left[\bar{V}^k\right] = \frac{1}{L}\mathbf{E}\left[f_2^k(\underline{h})\right] = p_1^k - p_2^k.$$

Now applying Corollary 7.1 of Theorem 7.1 to bound probability of deviations on both sides of the expected differences, let $t = t_k\sqrt{L}/L$, we have

$$\begin{aligned}
\mathbf{Pr}\left[\left|f_2^k(\underline{h}) - \mathbf{E}\left[f_2^k(\underline{h})\right]\right| \geq t_k\sqrt{L}\right] &= \\
&\mathbf{Pr}\left[\left|\bar{U}^k - \bar{V}^k - (\mathbf{E}\left[\bar{U}^k\right] - \mathbf{E}\left[\bar{V}^k\right])\right| \geq t_k\sqrt{L}/L\right] \\
\leq\ & 2e^{\frac{-2(t_k\sqrt{L}/L)^2}{(2/L)}} \\
\leq\ & 2e^{-t_k^2}.
\end{aligned}$$

□

Lemma 9.8. *In probability space $(\Omega, \mathcal{F}, \mathbf{Pr})$, for each balanced cut $(S, \bar{S})$,*

$$\mathbf{Pr}\left[\underline{h} \in \mathcal{E}_2^L\right] \leq \rho_2,$$

where $\rho_2 = O(\frac{1}{2^{2N}\mathrm{poly}(N)})$ and $N \geq 2$.

Proof. The proof follows that of Lemma 8.4, except with the following modifications: (9.4.29) is replaced with the following: Let $r_k, \forall k$ represent the number of such intervals: we have $\forall k$, so long as $N \geq 2$,

$$r_k = \log(\left|\sqrt{L}\right| + \left|L(p_1^k - p_2^k)/\sqrt{L}\right|) \tag{9.4.29}$$

$$\leq \log 2\sqrt{L} \leq \log 2\sqrt{N/2} \leq \log N. \tag{9.4.30}$$

We also modify Lemma 8.5 to the following: whose proof follows a similar line, except that the definition of Δ here follows that of Definition 9.13 due to the change of Lemma 9.7 as compared to Lemma 8.1.

Lemma 9.9. *Let $\Delta/4 = 2N\ln 2 + K(\ln 2)(\log\log N + 1) + (3\ln N)/8$ as Δ is defined in Definition 9.13.*

$$\mathbf{Pr}\left[\underline{h} \textit{ maps to a fixed } \mathbf{B}(\beta_1,\ldots,\beta_k) \textit{ s.t.} \sum_{k=1}^{K} \tilde{t}_k^2 \geq \Delta/4\right] \leq \frac{1}{2^{2N}\cdot(\log N)^K \cdot N^{3/2}}.$$

□

9.4.3 Bounded Differences

We are now ready to use bounded differences approach in $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$ and prove the following variation of Theorem 8.3.

Theorem 9.4. *Let $\underline{h}$ be a possible $2KL$-history that we record for a balanced cut $(S,\bar{S})$ such that $\underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L$. Then, for $t > 0$, in probability space $(\Omega_{\underline{h}}, \Sigma(\Omega_{\underline{h}}), \mathbf{Pr}_{\underline{h}})$, where all future $2(N-L)K$ random bits $\bar{f}$ are completely at random,*

$$\mathbf{Pr}_{\underline{h}}\left[\left|\mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{2KN}\right] - \mathbf{E}_{\underline{h}}\left[\mathit{diff}(\mathcal{T},(S,\bar{S}),L)\right]\right| \geq t\right] \leq 2e^{-t^2/2\sigma^2},$$

where $\sigma^2 \leq 4(N-L)L^2(K\gamma) + 4(N-L)L\Delta$, for all balanced cut $(S,\bar{S})$ with $0 < L \leq N/2$ swapped nodes.

Proof. We should substitute all mentioning of "a pair of bits" with a single bit; in particular, we substitute X_i^k, Y_i^k, wherever they are used in the proof of Theorem 8.3, with x_i^k, y_i^k to refer to a single bit at dimension k of point X and Y respectively.

In particular, we have

$$d_{i,k}(X_i) = \left|\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{(j)},x_i^k\right] - \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{(j)}\right]\right|. \quad (9.4.31)$$

And similarly, let $\ell' = 2KL + (N-L)K + (i-1)K + k - 1$, we have

$$d_{i,k}(Y_i) = \left|\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{(\ell')},y_i^k\right] - \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{(\ell')}\right]\right|.$$

We immediately have the following lemma that we can plug into Azuma's inequality, where $d_{i,k}$ applies to both $d_{i,k}(X_i)$ and $d_{i,k}(Y_i)$.

Lemma 9.10. *For the $2(N-L)K$ random bits on unswapped nodes X_i, Y_i $\forall i \in [1, N-L]$ that we reveal, at dimension $k \in [1,K]$, we have*

$$d_{i,k} \leq \left|L(p_2^k - p_1^k)\right| + \left|t_k\sqrt{L}\right|,$$

where t_k is defined in Definition 9.12 and Δ as in Definition 9.13, and $\sum_{k=1}^{K} t_k^2 \leq \Delta$.

Proof. This proof follows that of Lemma 8.8. Given that $Y_i, \forall i$, comes from $\mathbf{D}_2$ and $X_i, \forall i$, comes from $\mathbf{D}_1$, and by definition of $d_{i,k}(Y_i)$ and $d_{i,k}(X_i)$,

$$d_{i,k}(Y_i) = \begin{cases} \left|p_2^k\right|\left|f_2^k(\underline{h})\right| & : \quad y_i^k = 0, \\ \left|1-p_2^k\right|\left|f_2^k(\underline{h})\right| & : \quad y_i^k = 1, \end{cases}$$

and

$$d_{i,k}(X_i) = \begin{cases} \left|p_1^k\right|\left|f_2^k(\underline{h})\right| & : \quad x_i^k = 0, \\ \left|1-p_1^k\right|\left|f_2^k(\underline{h})\right| & : \quad x_i^k = 1. \end{cases}$$

Hence given that $\underline{h} \in \bar{\mathcal{E}}_2^L$, Lemma 9.6, and $\left|\mathbf{E}\left[f_2^k(\underline{h})\right]\right| = \left|L(p_2^k - p_1^k)\right|$ as in Proposition 9.6,

$$\begin{aligned} d_{i,k}(Y_i) &\leq \left|f_2^k(\underline{h})\right| &(9.4.32)\\ &\leq \left|\mathbf{E}\left[f_2^k(\underline{h})\right]\right| + \left|t_k\sqrt{L}\right| &(9.4.33)\\ &= \left|L(p_2^k - p_1^k)\right| + \left|t_k\sqrt{L}\right|, &(9.4.34)\end{aligned}$$

and similarly, $d_{i,k}(X_i) \leq \left|L(p_2^k - p_1^k)\right| + \left|t_k\sqrt{L}\right|$, where $\sum_{k=1}^K t_k^2 \leq \Delta$. □

We are now ready to obtain a bound for $\sigma^2 = 2\sum_{i=1}^{N-L}\sum_{k=1}^K d_{i,k}^2$, where $d_{i,k}^2 \leq \left|L(p_2^k - p_1^k)\right| + \left|\sqrt{L}(t_k)\right|)^2$ applies to unswapped nodes $X_i, Y_i, \forall i = 1, \ldots, N-L$, in bounding the differences they cause by revealing the random bits on dimension K.

Given that $\sum_{k=1}^K t_k^2 \leq \Delta$,

$$\begin{aligned} \sigma^2 &= \sum_{i,k}(d_{i,k}^2(X_i) + d_{i,k}^2(Y_i)) = 2\sum_{i,k} d_{i,k}^2 \\ &\leq 2\sum_{i=1}^{N-L}\sum_{k=1}^{K}\left(\left|L(p_2^k - p_1^k)\right| + \left|\sqrt{L}(t_k)\right|\right)^2 \\ &\leq 2(N-L)\sum_k 2(L(p_2^k - p_1^k))^2 + 2(\sqrt{L}(t_k))^2 \\ &= 4L^2(N-L)\sum_k (p_2^k - p_1^k)^2 + 4L(N-L)\sum_k t_k^2 \\ &\leq 4(N-L)L^2(K\gamma) + 4(N-L)L\Delta, \end{aligned}$$

where $\Delta = 8N\ln 2 + 4K\ln 2(\log\log N + 1) + 3\ln N/2$ as in Definition 9.13. □

We now apply Theorem 9.4 to obtain the following bound on a bad event. Note that the constant in the following lemma has not been optimized.

Lemma 9.11. *Let $\underline{h}$ be the specific $2KL$-history that we record for a balanced cut $(S, \bar{S})$ such that $\underline{h} \in \bar{\mathcal{E}}_1^L \cap \bar{\mathcal{E}}_2^L$. Let $\rho_3^L = \frac{2}{N^{4L}}$. Then*

$$\mathbf{Pr}\left[\mathit{diff}(\mathcal{T}, (S, \bar{S}), L) \leq 0 | \underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L, \bar{f} \text{ at random}\right] \leq \rho_3^L,$$

given that $K = \Omega(\frac{\ln N}{\gamma})$ and $KN = \Omega(\frac{\ln N \log\log N}{\gamma^2})$, for all $N \geq 4$.

Proof. We take $t = \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)\right] \geq KL(N-L)\gamma/2$ and plug in Theorem 9.4, we have the following:

$$\begin{aligned}
& \mathbf{Pr}\left[\mathsf{diff}(\mathcal{T},(S,\overline{S}),L) \leq 0 | \underline{h} \in \bar{\mathcal{E}}_2^L \cap \bar{\mathcal{E}}_1^L\right] \\
= \; & \mathbf{Pr}_{\underline{h}}\left[\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)|\underline{H}^{2KN}\right] - \mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)\right] \leq -\mathbf{E}_{\underline{h}}\left[\mathsf{diff}(\mathcal{T},(S,\bar{S}),L)\right]\right] \\
\leq \; & 2e^{-t^2/2\sigma^2} \leq 2e^{-(KL(N-L)\gamma/2)^2/2\sigma^2},
\end{aligned} \tag{9.4.35}$$

where σ^2 is defined in Theorem 8.3.

We first rewrite σ^2,

$$\begin{aligned}
\sigma^2 & \leq 4(N-L)L^2(K\gamma) + 4(N-L)L\Delta \\
& = 4(N-L)L^2(K\gamma) + (N-L)L\Lambda,
\end{aligned} \tag{9.4.36}$$

which is exactly $1/16$ of what we have in proof of Lemma 8.9, as in (8.5.55), where $\Lambda = 4\Delta$. Given that $t^2 = (KL(N-L)\gamma)^2/4$ is also $1/16$ of that in (8.5.54), the rest of the calculation is exactly the same as that in proof of Lemma 8.9, where $N \geq 4$ is assumed. □

We finish this chapter by noting that, to prove Theorem 9.1, we need to bundle the modified pieces with some original proofs in Chapter 8 according to the outline shown in Section 8.3.

10 Conclusions and Open Problems

10.1 Hierarchical Routing

As our work is motivated by routing and distributed data location applications in large-scale distributed systems, where hierarchy is introduced to address scalability issues, it is important to fully understand other issues such as robustness and dynamics posed by such applications, model them appropriately, and adapt our algorithms accordingly.

- In particular, how can we adapt the randomized constructions to an online setting, where nodes are allowed to join and leave the system dynamically, while still guaranteeing the bound on path stretch and the "optimality" of the decompositions?

- We also lack a complete understanding of the relationships between routing that optimizes path stretch versus routing that optimizes congestion. How can we effectively compromise these two goals?

10.2 EDP and Congestion Minimization

In EDPwC, the goal is to connect as many terminal pairs as possible subject to the constraint that at most ω demands can be routed through any edge in the graph. Note that when $\omega = O(\log n / \log\log n)$, we get a constant approximation via randomized rounding Raghavan and Thompson [1987]. For an undirected graph, the strongest hardness of approximation bounds are $\Omega(\log^{\frac{1}{2}-\varepsilon} n)$ for EDP and $\Omega(\log^{\frac{1-\varepsilon}{\omega+1}} n)$ for EDPwC due to Andrews et al. [2005].

(1) Is it possible to improve the hardness of approximation for undirected EDP to, e.g., $\Omega(\log^{1-\varepsilon} n)$?

(2) Is there a better approximation for All or Nothing Multicommodity Flow (ANF) problem such that the approximation ratio is $O(\log n)$?

(3) Is it possible to extend our algorithm to obtain a polylogarithmic approximation for undirected EDP in general graphs, in particular, when we allow congestion ω to grow from 1 to perhaps $\log\log n$?

10.3 Classification

In the context of this population classification problem, there are many open questions that one needs to consider to come up with rigorous proofs:

(1) How to extend this to biased cases, where two populations have different sizes? Currently, the max-cut theorem only works for balanced cases and the proof techniques strongly rely on the fact that we compare only all balanced cuts with the perfect partition.

(2) How to extend this analysis to multiple populations?

(3) How to allow admixture model, where each individual does not come from the same population of origin, instead, each individual's genotype can be a mixture of several distributions?

(4) How to analyze it when different loci are allowed to have correlations? Our current model assumes independence between different loci, and our program draws samples from the genotype databases randomly to simulate this independence.

Answering these questions will not only shed lights on our understanding of the underlying mathematical structure of DNA, but also have significance in defining and answering problems in this the classic domain of learning distributions.

Bibliography

ACHLIOPTAS, D. AND MCSHERRY, F. 2005. On spectral learning of mixtures of distributions. In *Proceedings of the 18th Annual Conference on Learning Theory*. 458–469. 8, 9

ALON, N. AND SPENCER, J. 1992. *The Probabilistic Method*. Wiley Interscience, New York. 26, 123

ANDREWS, M., CHUZHOY, J., KHANNA, S., AND ZHANG, L. 2005. Hardness of the undirected edge-disjoint paths problem with congestion. In *Proceedings of the 46th IEEE FOCS*. 59, 60, 173

ANDREWS, M. AND ZHANG, L. 2005a. Hardness of the undirected congestion minimization problem. In *Proceedings of the 37th ACM STOC*. 59, 60

ANDREWS, M. AND ZHANG, L. 2005b. Hardness of the undirected edge-disjoint path problem. In *Proceedings of the 37th ACM STOC*. 2

ANDREWS, M. AND ZHANG, L. 2006. Logarithmic hardness of the directed congestion minimization problem. In *Proceedings of the 38th ACM STOC*. 59, 60

ARORA, S. AND KANNAN, R. 2001. Learning mixtures of arbitrary gaussians. In *Proceedings of 33rd ACM Symposium on Theory of Computing*. 247–257. 8, 9

AUMANN, Y. AND RABANI, Y. 1995. Improved bounds for all-optical routing. In *Proceedings of the 6th ACM-SIAM SODA*. 567–576. 58

AWERBUCH, B., GAWLICK, R., LEIGHTON, F. T., AND RABANI, Y. 1994. On-line admission control and circuit routing for high performance computing and communication. In *Proceedings of the 35th IEEE FOCS*. 412–423. 58

AWERBUCH, B. AND PELEG, D. 1992. Routing with polynomial communication-space trade-off. *SIAM J. Discrete Math 5,* 2, 151–162. 14

BARTAL, Y. 1996. Probabilistic approximations of metric spaces and its algorithmic applications. In *Proceedings of the 37th IEEE FOCS*. 184–193. 18

BECK, J. 1991. An algorithmic approach to the Lovász local lemma. I. *Random Struct. Alg. 2,* 4, 343–365. 19, 26

BRODER, A., FRIEZE, A., AND UPFAL, E. 1994. Existence and construction of edge-disjoint paths on expander graphs. *SIAM Journal of Computing 23*, 976–989. 58

CHEKURI, C. AND KHANNA, S. 2003. Edge disjoint paths revisited. In *Proceedings of the 14th ACM-SIAM SODA*. 59, 60

CHEKURI, C., KHANNA, S., AND SHEPHERD, F. B. 2004a. The all-or-nothing multicommodity flow problem. In *Proceedings of the 36th ACM STOC*. 59, 92

CHEKURI, C., KHANNA, S., AND SHEPHERD, F. B. 2004b. Edge-disjoint paths in planar graphs. In *Proceedings of the 45th IEEE FOCS*. 59

CHEKURI, C., KHANNA, S., AND SHEPHERD, F. B. 2005. Multicommodity flow, well-linked terminals, and routing problems. In *Proceedings of the 37th ACM STOC*. 57, 59, 60, 61, 62, 63, 65, 66, 67

CHEKURI, C., KHANNA, S., AND SHEPHERD, F. B. 2006a. Edge-disjoint paths in planar graphs with constant congestion. In *Proceedings of the 38th ACM STOC*. 59

CHEKURI, C., KHANNA, S., AND SHEPHERD, F. B. 2006b. An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow. *Journal of Theory of Computing 2*, 137–146. 59, 60

CHERNOFF, H. 1952. A measure of asymptotic efficiency of tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics 23*, 493–507. 90

CHUNG, F. AND LU, L. 2006. Concentration inequalities and martingale inequalities — a survey. to appear. 122, 123

CHUZHOY, J. AND NAOR, J. 2004. New hardness results for congestion minimization and machine scheduling. In *Proceedings of the 36th ACM STOC*. 28–34. 59

COWEN, L. J. 2001. Compact routing with minimum stretch. *J. Algorithms 38,* 1, 170–183. 14

CRYAN, M. 1999. Learning and approximation algorithms for problems motivated by evolutionary trees. Ph.D. thesis, University of Warwick. 9

CRYAN, M., GOLDBERG, L., AND GOLDBERG, P. 2002. Evolutionary trees can be learned in polynomial time in the two state general markov model. *SIAM Journal of Computing 31,* 2, 375–397. 9

DASGUPTA, A., HOPCROFT, J., KLEINBERG, J., AND SANDLER, M. 2005. On learning mixtures of heavy-tailed distributions. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science*. 491–500. 8

DASGUPTA, S. 1999. Learning mixtures of gaussians. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer S cience*. 634–644. 8

DASGUPTA, S. AND SCHULMAN, L. J. 2000. A two-round variant of em for gaussian mixtures. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI)*. 8, 9

DEZA, M. M. AND LAURENT, M. 1997. *Geometry of cuts and metrics*. Springer-Verlag, Berlin. 16

FELDMAN, J., O'DONNELL, R., AND SERVEDIO, R. 2005. Learning mixtures of product distributions over discrete domains. In *Proceedings of the 46th IEEE FOCS*. 9

FELDMAN, J., O'DONNELL, R., AND SERVEDIO, R. 2006. Pac learning mixtures of gaussians with no separation assumption. In *Proceedings of the 19th Annual COLT*. 9

FREDERICKSON, G. N. AND JANARDAN, R. 1988. Designing networks with compact routing tables. *Algorithmica 3*, 171–190. 14

FREDERICKSON, G. N. AND JANARDAN, R. 1989. Efficient message routing in planar networks. *SICOMP 18,* 4, 843–857. 14

FREUND, Y. AND MANSOUR, Y. 1999. Estimating a mixture of two product distributions. In *Proceedings of the 12th Annual COLT*. 183–192. 9, 161

GARG, N., VAZIRANI, V. V., AND YANNAKAKIS, M. 1993. Primal-dual approximation algorithms for integral flow and multicut in trees. In *Proc. of the 20th ICALP*. 58

GARG, N., VAZIRANI, V. V., AND YANNAKAKIS, M. 1996. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM J. of Computing 25*, 235–251. 98

GAVOILLE, C. 2001. Routing in distributed networks: Overview and open problems. *ACM SIGACT News 32,* 1, 36–52. 14

GAVOILLE, C. AND GENGLER, M. 2001. Space-efficiency for routing schemes of stretch factor three. *JPDC 61,* 5, 679–687. 15

GUPTA, A., KRAUTHGAMER, R., AND LEE, J. R. 2003. Bounded geometries, fractals, and low–distortion embeddings. In *Proceedings of the 44th IEEE FOCS*. 534–543. 13, 14, 15, 16, 18, 20, 24

GURUSWAMI, V., KHANNA, S., RAJARAMAN, R., SHEPHERD, F. B., AND YANNAKAKIS, M. 1999. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. In *Proceedings of the 31th ACM STOC*. 59, 60

HEINONEN, J. 2001. *Lectures on analysis on metric spaces*. Springer-Verlag, New York. 16

HILDRUM, K., KUBIATOWICZ, J. D., RAO, S., AND ZHAO, B. Y. 2002. Distributed object location in a dynamic network. In *Proceedings of SPAA*. 41–52. 1, 13, 15, 35

HOEFFDING, W. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association 58,* 301, 13–30. 107

KANNAN, R., SALMASIAN, H., AND VEMPALA, S. 2005. The spectral method for general mixture models. In *Proc. of the 18th Annual COLT*. 8, 9

KARGER, D. R. 1994. Random sampling in cut, flow, and network design problems. In *Proceedings of the 26th ACM STOC*. 58, 88, 89

KARGER, D. R. 1999. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research 24,* 2, 383–413. 89, 90, 91

KARGER, D. R. AND RUHL, M. 2002. Finding nearest neighbors in growth-restricted metrics. In *Proceedings of the 34th ACM STOC*. 63–66. 1, 13, 15

KEARNS, M., MANSOUR, Y., RON, D., RUBINFELD, R., SCHAPIR, R., AND SELLIE, L. 1994. On the learnability of discrete distributions. In *Proceedings of the 26th ACM STOC*. 273–282. 9

KHANDEKAR, R., RAO, S., AND VAZIRANI, U. 2006. Graph partitioning using single commodity flows. In *Proceedings of the 38th ACM STOC*. 57, 93, 94

KLEINBERG, J. 2005. An approximation algorithm for the disjoint paths problem in even-degree planar graphs. In *Proceedings of the 46th IEEE FOCS*. 3, 59

KLEINBERG, J. AND RUBINFELD, R. 1996. Short paths in expander graphs. In *Proceedings of the 37th IEEE FOCS*. 58

KLEINBERG, J. AND TARDOS, E. 1995a. Approximations for the disjoint paths problem in high-diameter planar networks. In *Proceedings of the 27th ACM STOC*. 58

KLEINBERG, J. AND TARDOS, E. 1995b. Disjoint paths in densely embedded graphs. In *Proceedings of the 36th IEEE FOCS*. 52–61. 59

KLEINBERG, J. M. 1996. Approximation algorithms for disjoint paths problems. Ph.D. thesis, MIT, Cambridge, MA. 59, 60

KLEINROCK, L. AND KAMOUN, F. 1977. Hierarchical routing for large networks. Performance evaluation and optimization. *Computer Networks 1,* 3, 155–174. 1, 13, 15, 35

KOLLIOPOULOS, S. G. AND STEIN, C. 1998. Approximating disjoint-path problems using greedy algorithms and packing integer programs. In *Proceedings of IPCO*. 59, 60

KOLMAN, P. AND SCHEIDELER, C. 2001. Simple on-line algorithms for the maximum disjoint paths problem. In *Proceedings of the 13th ACM SPAA*. 58

KRAUTHGAMER, R. AND LEE, J. R. 2003. The intrinsic dimensionality of graphs. In *Proceedings of the 35th ACM STOC*. 438–447. 14

MOSSEL, E. AND ROCH, S. 2005. Learning nonsinglar phylogenies and hidden markov models. In *Proceedings of the 37th ACM STOC*. 9

MOTWANI, R. AND RAGHAVAN, P. 1995. *Randomized Algorithms*. Cambridge University Press, New York. 122

OBATA, K. 2004. Approximate max-integral-flow/min-multicut theorems. In *Proceedings of the 36th ACM STOC*. 58

PELEG, D. 2000. *Distributed Computing*. SIAM, Phila., PA. 14

PELEG, D. AND UPFAL, E. 1989. A trade-off between space and efficiency for routing tables. *JACM 36,* 3, 510–530. 14

PLAXTON, C. G., RAJARAMAN, R., AND RICHA, A. W. 1999. Accessing nearby copies of replicated objects in a distributed environment. *Theory Comput. Syst. 32,* 3, 241–280. 1, 13, 15

PRITCHARD, J. K., STEPHENS, M., AND DONNELLY, P. 2000. Inference of population structure using multilocus genotype data. *Genetics 155*, 954–959. 3, 4

RAGHAVAN, P. AND THOMPSON, C. D. 1987. Randomized roundings: a technique for provably good algorithms and algorithms proofs. *Combinatorica 7*, 365–374. 59, 60, 173

RAJARAMAN, R., RICHA, A. W., VOCKING, B., AND VUPPULURI, G. 2001. A data tracking scheme for general networks. In *Proceedings of SPAA*. 247–254. 15

ROBERTSON, N. AND SEYMOUR, P. D. 1990. An outline of a disjoint paths algorithm. *Paths, Flows and VLSI-design, Algorithms and Combinatorics 9*, 267–292. 2

SRINIVASAN, A. 1997. Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems. In *Proceedings of the 38th IEEE FOCS*. 59, 60

TALWAR, K. 2004. Bypassing the embedding: Algorithms for low-dimensional metrics. In *Proceedings of the 36th ACM STOC*. Chicago, IL, 281–290. 13, 15, 20, 24

VARADARAJAN, K. AND VENKATARAMAN, G. 2004. Graph decomposition and a greedy algorithm for edge-disjoint paths. In *Proceedings of the ACM-SIAM SODA*. 59, 60

VEMPALA, V. AND WANG, G. 2002. A spectral algorithm of learning mixtures of distributions. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*. 113–123. 8