ABSTRACT

Title of Thesis: Well-Separation Heuristics for the Metric Traveling Salesman Problem

Addison Logan Hanrattie
Master of Science, 2026

Thesis Directed by: Professor Hanan Samet
Department of Computer Science

Well-separated pair decompositions (WSPDs) are a fundamental tool in computational geometry and underlie many geometric approximation algorithms. Despite their widespread use, comparatively little is known about how well-separation constrains the structure of tours in the metric Traveling Salesman Problem (TSP).

In this work, we investigate the relationship between WSPDs and TSP tours, providing new insights into how well-separation can be leveraged to design efficient heuristics for the metric TSP. Our main result shows that if a point set can be partitioned into two well-separated pairs, then any optimal TSP tour must visit the points in each set consecutively. We then extend this result to the case of the multiple Traveling Salesman Problem where we prove that similar results hold for each configuration of endpoints. Furthermore, we develop efficient algorithms for testing whether a given point set can be partitioned into well-separated sets, and how to leverage such a partitioning to efficiently compute optimal TSP tours. Finally, we develop a heuristic which can iteratively improve a solution by checking for faults in the tour over each pair in a WSPD.

Well-Separation Heuristics for the Metric Traveling Salesman Problem

by

Addison Logan Hanrattie

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2026

Advisory Committee:
Professor Hanan Samet, Chair/Advisor
Professor David Mount
Professor Alan Liu
Dr. Jagan Sankaranarayanan (Google)

© Copyright by
Addison Logan Hanrattie
2026

Dedication

To *Mom* and *Dad*.

Acknowledgments

In this thesis, I show that the best way to deliver a set of packages among a set of cities is to visit them one city at a time rather than skipping around (a result likely obvious to most readers but which can nevertheless fill a 100-page thesis). While this may be true of the Traveling Salesman Problem, it is seemingly a fact of life that our journeys take random detours and side trips, whose value far surpasses that of the easier path. I am grateful to have had the opportunity to take such a detour when, through chance, I found myself sharing an elevator with Prof. Hanan Samet. It was in this moment that my "tour" was permanently diverted from the path of a typical computer science student, and I will be forever grateful. Hanan's mentorship and guidance have been invaluable to me throughout my academic journey; whether through the courses I took with him, the research we did together, his insightful advice, or our casual conversations about life in his office, I have learned so much from him and am grateful for his support and encouragement. I cannot overstate the generosity of Hanan and his wife, Dr. Leila De Floriani, who have always been willing to chat no matter where they are on Earth. The greatest lesson I have learned from Hanan is that the journey is more important than the destination, and I am grateful to have had the opportunity to take such a journey with him as my guide.

I am also grateful to the rest of my committee, Prof. David Mount, Prof. Alan Liu, and Dr. Jagan Sankaranarayanan, for their support and guidance throughout this process. Each of them has been an integral part of my educational and academic journey, and I deeply appreciate their dedication and insights. I would especially like to thank Dr. Jagan Sankaranarayanan for his continuous mentorship — from his comments on this thesis to his essential insights during

our research meetings. My multidisciplinary education has been the product of many exemplary professors across a number of departments. I would also like to thank my early mentors, Mr. James Zambarano and Mrs. Rachel DiGioia, who helped nurture my passion for STEM. Furthermore, I am grateful to the many staff members and administrators at the University of Maryland who assisted me during my time here. They empowered me to become a leader on the campus I love and to bring positive change not only to the programs I have interacted with across campus but also to the university as a whole.

I would also like to thank my many childhood friends, whose consistent companionship and interest in tech have been a source of inspiration and motivation for me to pursue a career in computer science. My time at the University of Maryland would not have been the same without my many friends here who never failed to make me laugh, and the memories we have made together will be some of the most cherished of my life. I am especially grateful to the intelligent men and women of my professional technology fraternity, $K\Theta\Pi$, whose unique paths and perspectives have molded my own, and whose friendship has been a source of joy and support throughout my time at Maryland. This thesis would not have been possible without the support of my girlfriend, Shreeni Chikyala, whose own passion for computer science has bolstered my own and whose sacrifice and assistance have been indispensable to me throughout this process.

Most of all, I am grateful to my family and extended family, who have always been there for me no matter what. To my parents, Catherine and Clark Hanrattie, who have always pushed me to be the best version of myself, who have sacrificed to ensure I could pursue my dreams, and who have been my biggest supporters throughout this process. To my brothers, Reese and Preston Hanrattie, who have been the best siblings a brother could ask for and who impress me every day with their own accomplishments. To my grandparents, who have showered me with love and

instilled in me a love for learning, education, and the world around me. To my dog, Duke, who has been my loyal companion and most *vocal* supporter throughout this process.

This work would not have been possible without all of you. *Thank you.*

Table of Contents

Dedication ii

Acknowledgements iii

Table of Contents vi

List of Figures viii

Chapter 1: Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Contributions . . . 3

Chapter 2: Background 6
- 2.1 Notation . . . 6
- 2.2 The Metric Traveling Salesman Problem . . . 6
- 2.3 Well-Separated Pair Decompositions . . . 7

Chapter 3: Well-Separated Sets are Path/Tour Separated 11
- 3.1 Single Paths and Tours . . . 11
- 3.2 Multi Paths of Endpoint Pairs . . . 16
  - 3.2.1 Two Paths With Fixed Endpoint Pairings . . . 16
  - 3.2.2 Extension to k Paths of Fixed Endpoint Pairings . . . 27
- 3.3 Multi Paths of Fixed Start and End Sets . . . 30
  - 3.3.1 Two Entrances and Two Exits . . . 30
  - 3.3.2 Extension to k Paths of Fixed Start and End Sets . . . 38
- 3.4 Multi Paths of Fixed Endpoint Sets . . . 41
  - 3.4.1 Two Paths / Four Endpoints . . . 42
  - 3.4.2 Extension to k Paths / $2k$ Endpoints . . . 45
- 3.5 Multi Paths of Cycle Forming Predefined Endpoint Pairs . . . 49

Chapter 4: Multiple Well-Separated Sets are Path/Tour Separated 61
- 4.1 Three Well-Separated Sets are Path/Tour Separated . . . 61
- 4.2 k Well-Separated Sets are Path/Tour Separated . . . 73

Chapter 5: Partitioning Algorithms 76
- 5.1 Bipartitions . . . 77
- 5.2 Partitioning into k sets . . . 81

Chapter 6: Applications of Global Separation 84
- 6.1 TSPLIB . . . 84
- 6.2 Varying Separation Factor . . . 85
  - 6.2.1 Point Generation . . . 85

6.2.2 Verification on TSPLIB 87

Chapter 7: General Heuristics Given a WSPD **88**
7.1 Identification Algorithm 89
7.2 Methodology 92
7.3 Results 93
7.3.1 $S = 1.5$ 93
7.3.2 Varying Separation Factor 94
7.3.3 Ablation Study 95

Chapter 8: Conclusion **97**
8.1 Related Work 97
8.1.1 Well-Separated Pair Decomposition 97
8.1.2 Edge Elimination 98
8.1.3 Clustered TSP 98
8.1.4 ΔTSP Heuristics 100
8.2 Future Work 100

Bibliography **104**

List of Figures

2.1 An example of a WSPD. Pairs between single points are omitted for clarity. 7

3.1 The 2-opt swap involves replacing the red inter-cluster edges with the green intra-cluster edges. The WSP condition ensures that the green edges are shorter than the red edges, leading to a shorter overall path. 12
3.2 By choosing a large enough p value the sets will be s-well-separated however the second tour will be strictly longer than the optimal tour. 15
3.3 The 3-opt swap involves cutting out the foreign part of the left path which moves through B and inserts it into the right path which begins and ends in B. The WSP condition ensures that the new connections are shorter than the old connections, leading to a shorter overall path. 17
3.4 For $s < \frac{3}{2}$ the extra crossing setup (a) is cheaper than the fewer crossing setup (b). . 20
3.5 By performing a 2-opt swap across the paths we can move from the configuration in (a) to the configuration in (b). The WSP condition ensures that the new connections are shorter than the old connections, leading to a shorter overall solution. 31
3.6 Behavior of a tour through two small well-separated sets. The dashed lines represent foreign parts of the tour outside of $A \cup B$ and solid lines represent the edges of the tour that are exclusively within $A \cup B$. 50

4.1 The tour with one less cycle across the three sets must be shorter by the WSP condition. 62

6.1 Count of problems out of 10,000 generated that had more than two edges crossing the bipartition in the optimal TSP solution. 86

7.1 Results of the guaranteed repair heuristic for $s = 1.5$. 94
7.2 Count of tours with optimality gap reductions by separation factor s. 95
7.3 Scatter plot of each tour comparing flagged pair repair methods vs randomized pair repair methods. 96

Chapter 1: Introduction

1.1 Motivation

The Traveling Salesman Problem (TSP) is a classic optimization problem that has been extensively studied in computer science and operations research [DFJ54; App+06]. Given a set of points and pairwise distances between them, the goal of the TSP is to find the shortest possible tour that visits each point exactly once and returns to the starting point. Despite its simplistic formulation, the complexity of the TSP has challenged researchers since before the inception of Computer Science as a field.

While the problem is easy to state, it is known to be *NP-hard* in its general form, meaning to date no polynomial-time algorithm has been found to solve optimally for all cases. In many real-world scenarios, a constrained form of TSP is often encountered known as the Metric TSP (ΔTSP). In this version, all nodes are fully connected and the distances between nodes satisfy the triangle inequality. Then it follows that for any three nodes A, B, and C, the direct distance from A to C is no greater than the distance from A to B plus the distance from B to C; furthermore all distances are non-negative and symmetric. Many forms of travel throughout real-world networks, such as road networks, air travel, server connections, and many more often satisfy the definition of forming a metric space or very nearly do. Even with this constraint however the problem remains NP-hard. Furthermore while there do exist polynomial time approximation algorithms for specific

cases such as the Euclidean [Aro98; Mit96] and low doubling dimension [Tal04] metric spaces, none are known for ΔTSP displaying a significant gap in our understanding of the problem. The ΔTSP is known to be APX-hard [PY93] and the best known approximation algorithm is a 1.5-approximation [Chr22; KKG21].

A common misconception is that the Well-Separated Pair Decomposition (WSPD) only applies to Euclidean spaces; however, a decomposition can be built for any metric space. WSPDs are a powerful tool used across many areas of computational geometry and have been used to design efficient algorithms for a variety of problems, including nearest neighbor search, clustering, and spanner construction [CK95]. Conceptually, a typical WSPD construction provides a hierarchical decomposition of a point set that partitions the $\binom{n}{2}$ interactions between points into a reduced number of well-separated pairs of subsets (linear in n for many metric spaces). By representing the interactions between points in this way, WSPDs allow problems to be rephrased from issues over a collection of individual points to issues over a collection of pairs of subsets, which can often be solved more efficiently. When two sets are well separated, many distances between points in the sets appear approximately equal; thus, these structures yield numerous different geometric properties between their points that can be exploited to solve problems more efficiently.

It is seemingly a property of nature that when a system is created its points often naturally cluster into sets that are well-separated from each other. This property can be observed in many real-world scenarios where it is much easier for one to travel within a cluster of neighbors than to exit and travel to another. One can consider how people cluster into buildings within a city and then how cities are separated from each other by mountains, rivers, and other natural barriers. In the same light one could also consider how within cities there are many short running roads and how between cities there are long running highways. One could also consider how servers cluster

themselves into compute nodes, racks, and data centers, and how these clusters are separated from each other by degrading bandwidth and increasing latency. Even space itself seems to cluster, first into galaxies, then into solar systems, and then into planets, and how traveling within these clusters is much easier than traveling between them. And at the smallest scales of the universe, we can even consider how quarks cluster into protons and neutrons, and how these clusters are separated from each other by the strong nuclear force and how atoms cluster into molecules. Given the prevalence of well-separated clusters in real-world systems, it is natural to ask how the structure of these clusters can be exploited to solve navigation problems such as the TSP more efficiently.

Compared to the most popular literature, we take a unique approach to leveraging the properties of WSPDs. When applying WSPDs to solve a given problem, the most common approach is to construct as small a WSPD as possible and then use the size of the WSPD as the basis for offering a speedup over other approaches. In our work rather than requiring the total size of the WSPD to be small, we instead require that the WSPD has a maximalist structure that covers the entire problem at all hierarchical levels. It is in using this approach that we are able to derive new relations between WSPDs of a given problem and the structure of optimal TSP tours over that problem, which in turn allows us to derive new heuristics for solving the ΔTSP more efficiently. To the best of our knowledge no other work has attempted to derive heuristics for solving the ΔTSP by analyzing the structure of optimal tours over WSPDs.

1.2 Contributions

1. We begin by proving that if a point set can be partitioned into two well-separated sets, then any optimal TSP tour must respect the structure of the partitioning. We show this by first

examining behavior over Hamiltonian paths and then extend this result to TSP tours.

2. We then extend this result to the problem where there are multiple salesmen across a number of scenarios.

   (a) The case of fixed starting and ending points for each salesman, where there are n salesmen who must connect n fixed starting points to n fixed ending points in a bijective manner.

   (b) The case of each salesman selecting their own starting and ending points from two separate sets of n starting and n ending points, where each point must be used by exactly one salesman.

   (c) The case of n salesmen who must connect $2n$ pre-determined endpoints in any start/end order such that each endpoint is used by exactly one salesman.

   (d) The case of n salesmen who must connect $2n$ pre-determined endpoints such that if the endpoints were part of a greater tour the tour would remain valid under the selections made by the salesmen.

3. We then expand the first result to the case of three pairwise well-separated sets and then to the case of n pairwise well-separated sets, showing that in each case the structure of the partition must be respected by any optimal TSP tour.

4. In order to make our method practical, we offer efficient algorithms for testing whether a given point set can be partitioned into two well-separated sets and for testing whether a given point set can be partitioned into n pairwise well-separated sets. We analyze the runtime of these algorithms and examine how they leverage other geometric properties.

5. We demonstrate the relevance of our results by identifying TSPLIB instances that can be partitioned into well-separated sets. We then study lowering the requirements for the separation factor of the well-separated sets and determine that with high probability our results still apply.

6. Finally, we leverage our previous theorems to build a heuristic for repairing TSP tours over certain flagged pairs of the WSPD. We show that this heuristic is effective and run an ablation study to ensure it is the WSPD structure that is driving the improvements.

Chapter 2: Background

In this chapter, we will cover the background required for understanding the rest of this thesis.

2.1 Notation

Definition 2.1.1. $\delta(Q, R) = \min_{q \in Q, r \in R} |qr|$

2.2 The Metric Traveling Salesman Problem

A metric space is a set of points P together with a distance function $d : P \times P \to \mathbb{R}$ that satisfies the following properties for all $x, y, z \in P$:

1. Non-negativity: $d(x, y) \geq 0$
2. Identity of indiscernibles: $d(x, y) = 0$ if and only if $x = y$
3. Symmetry: $d(x, y) = d(y, x)$
4. Triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$

The Metric Traveling Salesman Problem (ΔTSP) is a specific version of the Traveling Salesman Problem where the distance function d satisfies the properties of a metric space. Given a set of points P and a distance function d that defines a metric space, the goal of the ΔTSP is to find

the shortest possible tour that visits each point in P exactly once and returns to the starting point. Formally, we want to find a permutation π of the points in P that minimizes the total distance of the tour, which can be expressed as:

$$\min_{\pi} \sum_{i=1}^{n-1} d(\pi(i), \pi(i+1)) + d(\pi(n), \pi(1))$$

where n is the number of points in P and $\pi(i)$ denotes the i-th point in the permutation π.

2.3 Well-Separated Pair Decompositions

A Well-Separated Pair Decomposition (WSPD) is a hierarchical decomposition that combines the properties of a Pair Decomposition and Well-Separated Pairs (WSPs). It is a powerful tool in computational geometry that allows us to efficiently represent and query the relationships between pairs of points in a metric space. An example of a WSPD is shown in fig. 2.1.

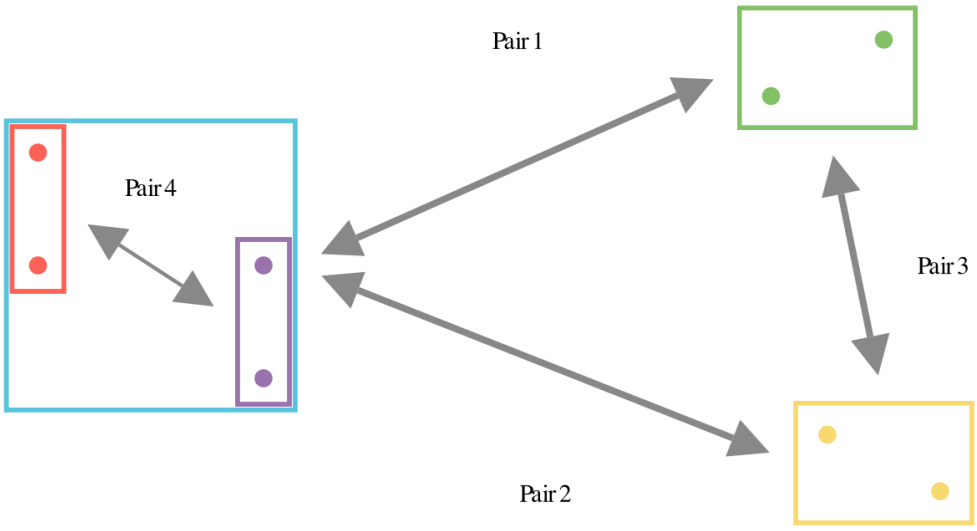


Figure 2.1: An example of a WSPD. Pairs between single points are omitted for clarity.

Definition 2.3.1. A **Pair Decomposition** of a point set P is a set of pairs

$$\mathcal{W} = \{\{A_1, B_1\}, \ldots, \{A_k, B_k\}\}$$

such that: (I) $\forall i; A_i, B_i \subset P$, (II) $\forall i; A_i \cap B_i = \emptyset$, and (III) $\bigcup_{i=1}^{k} (A_i \otimes B_i) = \binom{P}{2} = P \otimes P$. If $\forall p, q \in P, p \neq q \implies \exists! i; \{p, q\} \in A_i \otimes B_i$ then we say that $\mathcal{W}$ is a **Strict Pair Decomposition**.

Definition 2.3.2. The pair $\{Q, R\}$ (where $Q \cap R = \emptyset$) is s**-well-separated** if

$$\max\{\text{diam}(Q), \text{diam}(R)\} \leq \frac{1}{s} \cdot \delta(Q, R).$$

Corollary 2.3.2.1. *Given two sets of points A and B, there exists an s such that A and B are s-well-separated if and only if $\delta(A, B) > 0$.*

Definition 2.3.3. A Pair Decomposition $\mathcal{W}$ is a **Well-Separated Pair Decomposition** with separation factor s if $\forall \{A_i, B_i\} \in \mathcal{W}$, the pair $\{A_i, B_i\}$ is s-well-separated. If $\mathcal{W}$ is a Strict Pair Decomposition, then we say that $\mathcal{W}$ is a **Partition WSPD** otherwise it is a **Cover WSPD**.

We can then notice that if we have two well-separated sets of points, then the points in those sets must exhibit certain geometric properties.

Theorem 2.3.4 (The Fundamental Theorem of Well-Separated Pairs)**.** *Let A and B be two sets of points that are s-well-separated, let $p, p' \in A$, and let $q, q' \in B$. Then the following inequalities*

hold:

$$|pq| \leq (1 + 1/s) \cdot |pq'|$$

$$|pq| \leq (1 + 2/s) \cdot |p'q'|$$

$$|pp'| \leq (1/s) \cdot |pq|.$$

Proof. Let A and B be two sets of points that are s-well-separated, let $p, p' \in A$, and let $q, q' \in B$.

Note that $|qq'| \leq \mathrm{diam}(B) \leq \frac{1}{s} \cdot \delta(A, B) \leq \frac{1}{s}|pq'|$ by the definition of well-separated pairs. The first result then follows from the triangle inequality and above equation:

$$\begin{aligned} |pq| &\leq |pq'| + |q'q| \\ &\leq |pq'| + \frac{1}{s}|pq'| \\ &= (1 + 1/s) \cdot |pq'|. \end{aligned}$$

For the second result we note that $|pp'| \leq \mathrm{diam}(A) \leq \frac{1}{s} \cdot \delta(A, B) \leq \frac{1}{s}|p'q'|$ by the definition of well-separated pairs. The second result then follows from the triangle inequality and first equation:

$$\begin{aligned} |pq| &\leq |pp'| + |p'q'| + |q'q| \\ &\leq |pp'| + |p'q'| + \frac{1}{s}|p'q'| \\ &\leq \frac{1}{s}|p'q'| + |p'q'| + \frac{1}{s}|p'q'| \\ &= (1 + 2/s) \cdot |p'q'|. \end{aligned}$$

Finally for the third result we note that the following is true $|pp'| \leq \text{diam}(A) \leq \frac{1}{s} \cdot \delta(A, B) \leq \frac{1}{s}|pq|$ by the definition of well-separated pairs. The result is immediately apparent that $|pp'| \leq (1/s) \cdot |pq|$. □

Lastly, we can observe that the properties of well-separation transfer to any subsets of the well-separated sets.

Lemma 2.3.5. *Let A and B be two sets of points that are s-well-separated. Then, any $A' \subseteq A$ and $B' \subseteq B$ (where $|A'|, |B'| \geq 1$) are also s-well-separated.*

Proof. Let A and B be two sets of points that are s-well-separated and let $A' \subseteq A$ and $B' \subseteq B$ (where $|A'|, |B'| \geq 1$). We can then note that $\delta(A', B') \geq \delta(A, B)$ since $A' \subseteq A$ and $B' \subseteq B$. We can also note that $\text{diam}(A') \leq \text{diam}(A)$ and $\text{diam}(B') \leq \text{diam}(B)$ since $A' \subseteq A$ and $B' \subseteq B$. Thus we have the following:

$$\max\{\text{diam}(A'), \text{diam}(B')\} \leq \max\{\text{diam}(A), \text{diam}(B)\} \leq \frac{1}{s} \cdot \delta(A, B) \leq \frac{1}{s} \cdot \delta(A', B').$$

The result follows. □

Chapter 3: Well-Separated Sets are Path/Tour Separated

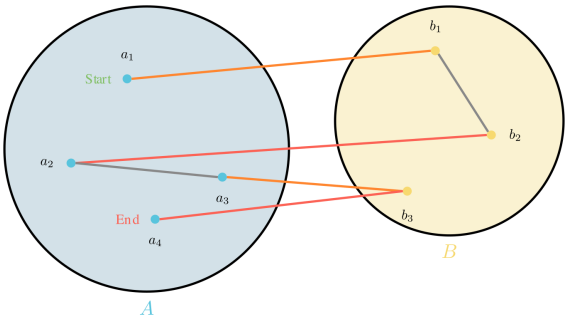
In this chapter, we will show that if we have two well-separated sets of points, then solutions to shortest path/tour problems over these points must respect the partition.

3.1 Single Paths and Tours

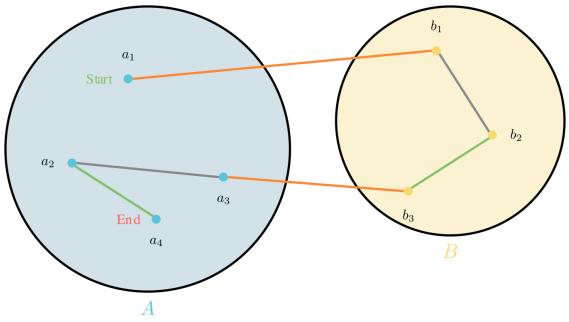
In the case of a single well-separated pair, we can show that if A and B are two sets of points that are s-well-separated and one endpoint is in A then all of the points in B must be visited consecutively in a shortest path/tour. This is a very powerful property as if $|A| \approx |B|$ then we can effectively halve the complexity of the problem by alternating over connections and then solving the two smaller problems. We will first show that this behaviour applies to the optimal Hamiltonian path with fixed endpoints — also known as the TSP path or open TSP problem with fixed endpoints. Once established, it trivially extends to the standard TSP.

To prove this claim, we employ a proof by contradiction that determines if an optimal tour did not consecutively visit points in A and B then we can find a 2-optimality violation by swapping costly jumps between A and B with cheaper connections within A and B. This 2-opt swap is illustrated in fig. 3.1.

Theorem 3.1.1. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$. Then there exists an optimal Hamiltonian path between endpoints $a' \in A$ and $b' \in B$ over $A \cup B$ that visits all points in A before any point in B. Furthermore if $s > 1$ then all optimal*



(a) An example of a path with extra crossings between A and B.



(b) A new path that avoids extra crossings between A and B.

Figure 3.1: The 2-opt swap involves replacing the red inter-cluster edges with the green intra-cluster edges. The WSP condition ensures that the green edges are shorter than the red edges, leading to a shorter overall path.

Hamiltonians must visit all points in A before any point in B.

Proof. Assume that for some optimal Hamiltonian path between endpoints $a' \in A$ and $b' \in B$ over $A \cup B$ it does not visit all points in A before any point in B. Then it must contain at least three edges crossing between A and B. Let these edges be $(a_1, b_1), (b_2, a_2), (a_3, b_3)$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to a_2 \rightsquigarrow a_3 \to b_3 \rightsquigarrow b'$$

Since the path is optimal we know that it satisfies the 2-optimality condition implying the following: $|a_1b_1| + |a_3b_3| \leq |a_1a_3| + |b_1b_3|$. This is to say that the following path must be at least as long as the original path

$$a' \rightsquigarrow a_1 \to a_3 \rightsquigarrow a_2 \to b_2 \rightsquigarrow b_1 \to b_3 \rightsquigarrow b'$$

Since A and B are s-well-separated we know that $\forall i, j, k; |a_ia_j| \leq \frac{1}{s}|a_ib_k|$ (and vice versa

for B). From this we can conclude the following:

$$\begin{aligned} |a_1b_1| + |a_3b_3| &\leq |a_1a_3| + |b_1b_3| \\ &\leq \frac{1}{s}|a_1b_1| + \frac{1}{s}|a_3b_3| \\ &= \frac{1}{s}(|a_1b_1| + |a_3b_3|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s > 1$ this is a contradiction and therefore any non-consecutive path cannot be optimal. If $s = 1$ then we can conclude that $|a_1b_1| + |a_3b_3| = |a_1a_3| + |b_1b_3|$ which implies that the newly constructed path is also optimal. Since the previous method strictly decreases the number of loops, it follows via induction that after repeated applications there must exist an optimal Hamiltonian path that visits all points in A before any point in B. □

Theorem 3.1.2. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$. Then there exists an optimal Hamiltonian path between endpoints $a', a'' \in A$ over $A \cup B$ that visits all points in B consecutively. Furthermore if $s > 1$ then all optimal Hamiltonian paths must visit all points in B consecutively.*

Proof. Assume that for some optimal Hamiltonian path between endpoints $a', a'' \in A$ over $A \cup B$ it does not visit all points in B consecutively. Then the path must contain at least four edges crossing between A and B. Let these edges be $(a_1, b_1), (b_2, a_2), (a_3, b_3), (b_4, a_4)$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to a_2 \rightsquigarrow a_3 \to b_3 \rightsquigarrow b_4 \to a_4 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 2-optimality condition implying the following: $|b_2a_2| + |b_4a_4| \leq |b_2b_4| + |a_2a_4|$. This is to say that the following path must be at least as long as the original path:

$$a' \rightsquigarrow a_1 \rightarrow b_1 \rightsquigarrow b_2 \rightarrow b_4 \rightsquigarrow b_3 \rightarrow a_3 \rightsquigarrow a_2 \rightarrow a_4 \rightsquigarrow a''$$

Since A and B are s-well-separated we know that $\forall i, j, k; |a_ia_j| \leq \frac{1}{s}|a_ib_k|$ (and vice versa for B). From this we can conclude the following:

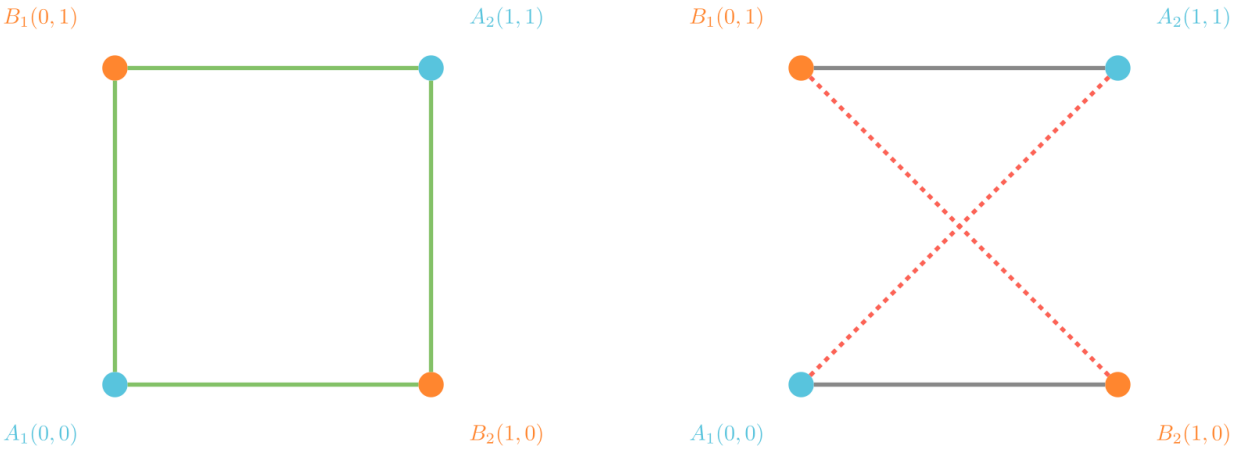
$$\begin{aligned} |b_2a_2| + |b_4a_4| &\leq |b_2b_4| + |a_2a_4| \\ &\leq \frac{1}{s}|b_2a_2| + \frac{1}{s}|b_4a_4| \\ &= \frac{1}{s}(|b_2a_2| + |b_4a_4|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s > 1$ this is a contradiction and therefore any non-consecutive path cannot be optimal. If $s = 1$ then we can conclude that $|b_2a_2| + |b_4a_4| = |b_2b_4| + |a_2a_4|$ which implies that the newly constructed path is also optimal. Since the previous method strictly decreases the number of loops, it follows via induction that after repeated applications there must exist an optimal Hamiltonian path that visits all points in B consecutively. □

Corollary 3.1.2.1. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$. Then there exists an optimal TSP tour on $A \cup B$ that visits all points in A before any point in B (or vice versa). Furthermore if $s > 1$ then all optimal TSP tours must satisfy this property.*

The bound on $s \geq 1$ is tight in the previous theorems; if $s < 1$ then we can construct

examples of well-separated pairs where the optimal path/tour does not visit all points in one set before any point in the other set. This can be demonstrated by placing the pairs at opposite edges of a unit square and using an L_p norm of sufficiently high p to make the separation factor appropriate and observing that while the pairs are s-well-separated crossing between the two sets would cause an intersection of two path segments — a condition that is impossible in an optimal (shortest) path/tour. The two possible tours are illustrated in fig. 3.2.



(a) The optimal tour with length 4.

(b) The consecutive tour with length $2+2(2^{1/p})$.

Figure 3.2: By choosing a large enough p value the sets will be s-well-separated however the second tour will be strictly longer than the optimal tour.

Remark. *The bound on s in Corollary 3.1.2.1 is tight.*

Proof. Fix any $s < 1$. Next we define $p > \max(1, -\frac{1}{\log_2(s)})$ and consider the L_p-norm on $\mathbb{R}^2$. We can then note that $s < 2^{-1/p}$. Now we can consider the following partition of the unit square $A = \{(0,0),(1,1)\}$ and $B = \{(0,1),(1,0)\}$.

The diameter of each set is the length of the diagonal of the unit square within the L_p-norm. This distance is equal to $(1^p + 1^p)^{1/p} = 2^{1/p}$. The distance between the two sets is one since the closest points between the two sets are one unit apart in either the x or y direction. Thus the

separation factor of these two sets is $\frac{1}{2^{1/p}} = 2^{-1/p}$. Since $s < 2^{-1/p}$ we know that the two sets are s-well-separated.

We can then note that the optimal TSP tour on $A \cup B$ has length 4 and must be $((0,0) \to (0,1) \to (1,1) \to (1,0) \to (0,0))$ or $((0,0) \to (1,0) \to (1,1) \to (0,1) \to (0,0))$ both of which visit points in A and B alternately. Any tour which visits all points in A before any point in B (or vice versa) must have length at least $2 \cdot 2^{1/p} + 2$ which is strictly greater than the optimal tour which has length 4 (since $0 < p < \infty$). □

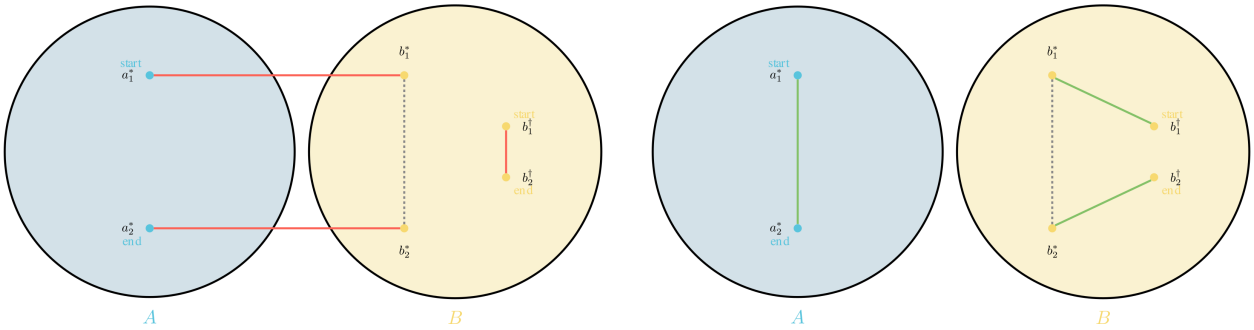
3.2 Multi Paths of Endpoint Pairs

The most restrictive case for multipath problems is one where the endpoints are fixed. In this case, we can show that if we have two well-separated sets of points and two paths with endpoints in each set then the optimal solution will have a specific structure. Specifically a path will either never cross between the two sets or it will cross between the two sets exactly once or twice depending on the location of its endpoints. In order to maintain the same endpoints for each path, the theorems require a more restrictive bound of $s \geq \frac{3}{2}$. However, once this bound is satisfied, the structure of the optimal solution remains invariant as s increases.

3.2.1 Two Paths With Fixed Endpoint Pairings

To begin, we will show that if we have two well-separated sets of points and two paths with endpoints in each set then the optimal solution will have a specific structure. Whereas in the previous section we leverage 2-opt swaps between crossing edges to fix them we cannot in a similar sense apply those swaps across paths as doing so would change their endpoints and thus violate the

problem constraints. Instead, the proofs here rely on showing that if the optimal solution lacked the required structure then we could cut the set-crossing edges from one path and a set-interior edge from the other path and reinsert the crossed part of the first path into the second path at the location of the set-interior edge to get a shorter tour while preserving the endpoints of the paths. An example of this swap is illustrated in fig. 3.3 where one path must start and end in A and the other path must start and end in B.



(a) The left path starts in A then costly crosses into B and then returns.

(b) The new paths that avoids extra crossings between A and B.

Figure 3.3: The 3-opt swap involves cutting out the foreign part of the left path which moves through B and inserts it into the right path which begins and ends in B. The WSP condition ensures that the new connections are shorter than the old connections, leading to a shorter overall path.

Lemma 3.2.1. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq \frac{3}{2}$. Then the multiple optimal Hamiltonian paths problem where one path had fixed endpoints $a', a'' \in A$ and the other path had fixed endpoints $a''', a'''' \in A$ has a solution where one of the path visits all points in B consecutively and the other path has no edges connecting A to B. Furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.*

Proof. Consider some optimal solution to the problem. Suppose that both paths have edges connecting A to B. Then there must be two edges of the form (a_1^*, b_1^*) and (b_2^*, a_2^*) in the first set and two edges of the form $(a_1^\dagger, b_1^\dagger)$ and $(b_2^\dagger, a_2^\dagger)$ in the second set where $a_1^*, a_2^*, a_1^\dagger, a_2^\dagger \in A$ and

$b_1^*, b_2^*, b_1^\dagger, b_2^\dagger \in B$. Thus these paths have the following partial structure:

$$a' \rightsquigarrow a_1^* \to b_1^* \rightsquigarrow b_2^* \to a_2^* \rightsquigarrow a'' \qquad a''' \rightsquigarrow a_1^\dagger \to b_1^\dagger \rightsquigarrow b_2^\dagger \to a_2^\dagger \rightsquigarrow a''''$$

Since the solution is optimal we know that they satisfy the 3-optimality condition implying the following: $|a_1^* b_1^*| + |b_2^* a_2^*| + |a_1^\dagger b_1^\dagger| \leq |a_1^* a_2^*| + |a_1^\dagger b_1^*| + |b_2^* b_1^\dagger|$. This is to say that the following paths must have a combined length at least as long as the original paths:

$$a' \rightsquigarrow a_1^* \to a_2^* \rightsquigarrow a'' \qquad a''' \rightsquigarrow a_1^\dagger \to b_1^* \rightsquigarrow b_2^* \to b_1^\dagger \rightsquigarrow b_2^\dagger \to a_2^\dagger \rightsquigarrow a''''$$

We can define L to be the combined length of the original paths and L' to be the combined length of the newly constructed paths. We can then note the following about the change in length between L and L': $L - L' = (|a_1^* b_1^*| + |b_2^* a_2^*| + |a_1^\dagger b_1^\dagger|) - (|a_1^* a_2^*| + |a_1^\dagger b_1^*| + |b_2^* b_1^\dagger|)$.

Define $\omega = \delta(A, B)$. Using the definitions of well separation we have:

$$|b_2^* a_2^*| \geq \omega \qquad |a_1^\dagger b_1^\dagger| \geq \omega \qquad |a_1^* a_2^*| \leq \frac{1}{s}\omega \qquad |a_1^\dagger a_1^*| \leq \frac{1}{s}\omega \qquad |b_2^* b_1^\dagger| \leq \frac{1}{s}\omega$$

The following can then be determined based on these inequalities and the properties of a

metric space:

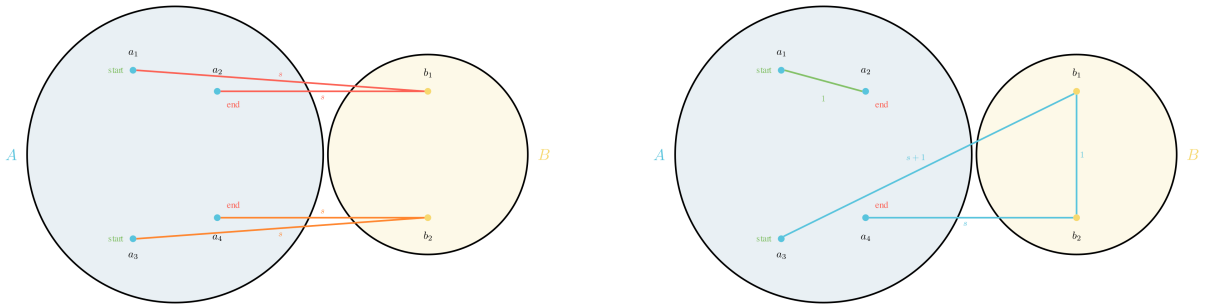
$$\begin{aligned}
L - L' &= (|a_1^* b_1^*| + |b_2^* a_2^*| + |a_1^\dagger b_1^\dagger|) - (|a_1^* a_2^*| + |a_1^\dagger b_1^*| + |b_2^* b_1^\dagger|) \\
&\geq (|a_1^* b_1^*| + |b_2^* a_2^*| + |a_1^\dagger b_1^\dagger|) - (|a_1^* a_2^*| + |a_1^\dagger a_1^*| + |a_1^* b_1^*| + |b_2^* b_1^\dagger|) \\
&= (|b_2^* a_2^*| + |a_1^\dagger b_1^\dagger|) - (|a_1^* a_2^*| + |a_1^\dagger a_1^*| + |b_2^* b_1^\dagger|) \\
&\geq 2\omega - (|a_1^* a_2^*| + |a_1^\dagger a_1^*| + |b_2^* b_1^\dagger|) \\
&\geq 2\omega - \frac{3}{s}\omega
\end{aligned}$$

If $s > \frac{3}{2}$ then we have that $L - L' > 0$ which is a contradiction since if the original path were optimal then we would have $L - L' \leq 0$. Therefore no optimal path can have both paths visiting both A and B if $s > \frac{3}{2}$. If $s = \frac{3}{2}$ then we have that $L - L' \geq 0$ which combined with $L - L' \leq 0$ (by optimality) implies that $L - L' = 0$. Therefore the newly constructed paths must also be optimal solutions to the problem. Since the number of crossings between A and B has strictly decreased we can apply this logic repeatedly to construct an optimal solution where only a single path has edges connecting A to B and the other path has no edges connecting A to B as desired.

We can then consider a case where only a single path has edges connecting A to B. We can then mark A' to be the set of points in A that are visited by this path and B' to be the set of points in B that are visited by this path. Then we can note that A' and B' are still s-well-separated by Lemma 2.3.5. We can then apply Theorem 3.1.2 to construct a solution where this path visits all points in B consecutively if it was not already the case. □

The bound on s in the previous theorem is tight; if $s < \frac{3}{2}$ then we can construct examples

of well-separated pairs where the optimal solution to the multiple optimal paths problem would require both paths to cross between the two sets (see fig. 3.4).



(a) Both paths cross into A and B for a cost of $4s$ combined.

(b) One path stays in A and the other path visits all of B for a cost of $2s + 3$ combined.

Figure 3.4: For $s < \frac{3}{2}$ the extra crossing setup (a) is cheaper than the fewer crossing setup (b).

Remark. *The bound on s in Lemma 3.2.1 is tight.*

Proof. Let $0.5 \leq s < \frac{3}{2}$. Consider two sets of points $A = \{a_1, a_2, a_3, a_4\}$ and $B = \{b_1, b_2\}$. Let $d(a_i, a_j) = 1$ for all $i \neq j$ and $d(b_1, b_2) = 1$. Let $d(a_1, b_1) = d(a_2, b_1) = d(a_3, b_2) = d(a_4, b_2) = s$. Let $d(a_1, b_2) = d(a_2, b_2) = d(a_3, b_1) = d(a_4, b_1) = s + 1$. Then A and B are s-well-separated since $\mathrm{diam}(A) = 1$, $\mathrm{diam}(B) = 1$ and $\delta(A, B) = s$.

Consider the multiple optimal Hamiltonian paths problem where one path has fixed endpoints a_1, a_2 and the other path has fixed endpoints a_3, a_4. Then we can note that there are six possible solutions to this problem:

- $P_1 = a_1 \to b_1 \to a_2$ and $P_2 = a_3 \to b_2 \to a_4$ with a total length of $4s$
- $P_1 = a_1 \to b_2 \to a_2$ and $P_2 = a_3 \to b_1 \to a_4$ with a total length of $4s + 4$
- $P_1 = a_1 \to a_2$ and $P_2 = a_3 \to b_1 \to b_2 \to a_4$ with a total length of $2s + 3$

- $P_1 = a_1 \to a_2$ and $P_2 = a_3 \to b_2 \to b_1 \to a_4$ with a total length of $2s + 3$
- $P_1 = a_1 \to b_1 \to b_2 \to a_2$ and $P_2 = a_3 \to a_4$ with a total length of $2s + 3$
- $P_1 = a_1 \to b_2 \to b_1 \to a_2$ and $P_2 = a_3 \to a_4$ with a total length of $2s + 3$

Since $s < \frac{3}{2}$ we have that $4s < 2s + 3$. Therefore no optimal solution can have one path visiting all points in B consecutively and the other path having no edges connecting A to B. □

Lemma 3.2.2. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq \frac{3}{2}$. Then the multiple optimal Hamiltonian paths problem where one path had fixed endpoints $a', a'' \in A$ and the other path had fixed endpoints $a''' \in A$ and $b' \in B$ has a solution where the first path visits only points in A and the second path visits all points in B consecutively. Furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.*

Proof. Begin by considering some optimal solution to the problem where the first path visits some points in B. Then we can note the following edges must exist in the first path (a_1^*, b_1^*) and (b_2^*, a_2^*) where $a_1^*, a_2^* \in A$ and $b_1^*, b_2^* \in B$ and in the second path there must be an edge of the form $(a^\dagger, b^\dagger)$ where $a^\dagger \in A$ and $b^\dagger \in B$. Thus these paths have the following partial structure:

$$a' \rightsquigarrow a_1^* \to b_1^* \rightsquigarrow b_2^* \to a_2^* \rightsquigarrow a'' \qquad a''' \rightsquigarrow a^\dagger \to b^\dagger \rightsquigarrow b'$$

Since the solution is optimal we know that they satisfy the 3-optimality condition implying the following: $|a_1^* b_1^*| + |b_2^* a_2^*| + |a^\dagger b^\dagger| \leq |a_1^* a_2^*| + |a^\dagger b_1^*| + |b_2^* b^\dagger|$. This is to say that the following paths must have a combined length at least as long as the original paths:

$$a' \rightsquigarrow a_1^* \to a_2^* \rightsquigarrow a'' \qquad a''' \rightsquigarrow a^\dagger \to b_1^* \rightsquigarrow b_2^* \to b^\dagger \rightsquigarrow b'$$

We can define L to be the combined length of the original paths and L' to be the combined length of the newly constructed paths. We can then note the following about the change in length between L and L': $L - L' = (|a_1^* b_1^*| + |b_2^* a_2^*| + |a^\dagger b^\dagger|) - (|a_1^* a_2^*| + |a^\dagger b_1^*| + |b_2^* b^\dagger|) \leq 0$ by optimality.

Define $\omega = \delta(A, B)$. Using the definitions of well separation and rules of a metric space we have:

$$|a_1^* b_1^*| \geq \omega \qquad |b_2^* a_2^*| \geq \omega \qquad |a_1^* a_2^*| \leq \frac{1}{s}\omega \qquad |b^\dagger b_1^*| \leq \frac{1}{s}\omega \qquad |b_2^* b^\dagger| \leq \frac{1}{s}\omega$$

The following can then be determined based on these inequalities and the properties of a metric space:

$$\begin{aligned}
L - L' &= (|a_1^* b_1^*| + |b_2^* a_2^*| + |a^\dagger b^\dagger|) - (|a_1^* a_2^*| + |a^\dagger b_1^*| + |b_2^* b^\dagger|) \\
&\geq (|a_1^* b_1^*| + |b_2^* a_2^*| + |a^\dagger b^\dagger|) - (|a_1^* a_2^*| + |a^\dagger b^\dagger| + |b^\dagger b_1^*| + |b_2^* b^\dagger|) \\
&= (|a_1^* b_1^*| + |b_2^* a_2^*|) - (|a_1^* a_2^*| + |b^\dagger b_1^*| + |b_2^* b^\dagger|) \\
&\geq 2\omega - (|a_1^* a_2^*| + |b^\dagger b_1^*| + |b_2^* b^\dagger|) \\
&\geq 2\omega - \frac{3}{s}\omega
\end{aligned}$$

If $s > \frac{3}{2}$ then we have that $L - L' > 0$ which is a contradiction since if the original path were optimal then we would have $L - L' \leq 0$. Therefore no optimal path can have both paths visiting both A and B if $s > \frac{3}{2}$. If $s = \frac{3}{2}$ then we have that $L - L' \geq 0$ which combined with $L - L' \leq 0$ (by optimality) implies that $L - L' = 0$. Therefore the newly constructed paths must also be optimal solutions to the problem. Since the number of crossings between A and B has strictly decreased

we can apply this logic repeatedly to construct an optimal solution where only the second path has edges connecting A to B.

We can then consider a case where only the second path has edges connecting A to B. We can then mark A' to be the set of points in A that are visited by this path and B' to be the set of points in B that are visited by this path. Then we can note that A' and B' are still s-well-separated by Lemma 2.3.5. We can then apply Theorem 3.1.1 to construct a solution where this path visits all points in B consecutively if it was not already the case. □

Lemma 3.2.3. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$. Then the multiple optimal Hamiltonian paths problem where one path had fixed endpoints $a' \in A$ and $b' \in B$ and the other path had fixed endpoints $a'' \in A$ and $b'' \in B$ has a solution where each path only has a single edge connecting A to B and furthermore if $s > 1$ then all optimal solutions must satisfy this property.*

Proof. Consider some optimal solution to the problem. We can define A' to be the set of points in A that are visited by the first path and B' to be the set of points in B that are visited by the first path. We can also define A'' to be the set of points in A that are visited by the second path and B'' to be the set of points in B that are visited by the second path. Then we can note that A' and B' are still s-well-separated and A'' and B'' are still s-well-separated by Lemma 2.3.5. We can then apply Theorem 3.1.1 to both of the paths which states that if $s > 1$ then the two paths could not have more than one edge connecting A to B and if $s = 1$ then there exists an optimal solution for each of the subproblems where they only cross A and B once. □

Lemma 3.2.4. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq \frac{3}{2}$. Then the multiple optimal Hamiltonian paths problem where one path had fixed endpoints $a', a'' \in$*

A and the other path had fixed endpoints $b', b'' \in B$ has a solution where neither path has an edge connecting A to B and furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.

Proof. Assume that for some optimal solution it has at least one edge connecting A to B. We will define P_A to be the first path that connects a' to a'' and P_B to be the second path that connects b' to b''. WLOG assume that P_A has an edge connecting A to B (if not then flip the names of A and B)

Case 1: P_B also traverses part of A. Therefore we have the following two edges in P_A: $(a_1^*, b_1^*), (b_2^*, a_2^*)$ and the following edges in P_B: $(b_1^\dagger, a_1^\dagger), (a_2^\dagger, b_2^\dagger)$ all ordered as they appear in their respective paths. This gives the following partial structure of the paths:

$$a'_s \rightsquigarrow a_1^* \to b_1^* \rightsquigarrow b_2^* \to a_2^* \rightsquigarrow a'_e \qquad b'_s \rightsquigarrow b_1^\dagger \to a_1^\dagger \rightsquigarrow a_2^\dagger \to b_2^\dagger \rightsquigarrow b'_e$$

Since the paths are optimal we know that they satisfy the 4-optimality condition implying the following: $|a_1^* b_1^*| + |b_2^* a_2^*| + |b_1^\dagger a_1^\dagger| + |a_2^\dagger b_2^\dagger| \leq |a_1^* a_2^\dagger| + |a_1^\dagger a_2^*| + |b_1^\dagger b_2^*| + |b_1^* b_2^\dagger|$. This is to say that the following paths must have a combined length at least as long as the original paths:

$$a'_s \rightsquigarrow a_1^* \to a_2^\dagger \rightsquigarrow a_1^\dagger \to a_2^* \rightsquigarrow a'_e \qquad b'_s \rightsquigarrow b_1^\dagger \to b_2^* \rightsquigarrow b_1^* \to b_2^\dagger \rightsquigarrow b'_e$$

Since A and B are s-well-separated we know that $\forall i, j, k; |a_i a_j| \leq \frac{1}{s}|a_i b_k|$ (and vice versa for B). From this we can conclude the following:

$$\begin{aligned}
|a_1^* b_1^*| + |b_2^* a_2^*| + |b_1^\dagger a_1^\dagger| + |a_2^\dagger b_2^\dagger| &\leq |a_1^* a_2^\dagger| + |a_1^\dagger a_2^*| + |b_1^\dagger b_2^*| + |b_1^* b_2^\dagger| \\
&\leq \frac{1}{s}|a_1^* b_1^*| + \frac{1}{s}|b_2^* a_2^*| + \frac{1}{s}|b_1^\dagger a_1^\dagger| + \frac{1}{s}|a_2^\dagger b_2^\dagger| \\
&= \frac{1}{s}(|a_1^* b_1^*| + |b_2^* a_2^*| + |b_1^\dagger a_1^\dagger| + |a_2^\dagger b_2^\dagger|)
\end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. This however is a contradiction since we assumed that $s \geq \frac{3}{2}$.

Case 2: P_B does not traverse any part of A. Then we must have two edges in P_A: $(a_1^*, b_1^*), (b_2^*, a_2^*)$ and one edge in P_B: $(b_1^\dagger, b_2^\dagger)$ all ordered as they appear in their respective paths. Furthermore it follows from Theorem 3.1.2 that these must be the only two edges between A and B. This gives the following partial structure of the paths:

$$a'_s \rightsquigarrow a_1^* \to b_1^* \rightsquigarrow b_2^* \to a_2^* \rightsquigarrow a'_e \qquad b'_s \rightsquigarrow b_1^\dagger \to b_2^\dagger \rightsquigarrow b'_e$$

Since the paths are optimal we know that they satisfy the 3-optimality condition implying the following: $|a_1^* b_1^*| + |b_2^* a_2^*| + |b_1^\dagger b_2^\dagger| \leq |a_1^* a_2^*| + |b_1^\dagger b_1^*| + |b_2^* b_2^\dagger|$ This is to say that the following paths must have a combined length at least as long as the original paths:

$$a'_s \rightsquigarrow a_1^* \to a_2^* \rightsquigarrow a'_e \qquad b'_s \rightsquigarrow b_1^\dagger \to b_1^* \rightsquigarrow b_2^* \to b_2^\dagger \rightsquigarrow b'_e$$

We can define L to be the total length of the two paths as is. We can then build a new solution by shortcutting the path by directly connecting a_1^* to a_2^* within P_A to form P'_A and we can break the edge $(b_1^\dagger, b_2^\dagger)$ in P_B and then connect $b_1^\dagger$ to b_1^* and b_2^* to $b_2^\dagger$ to form P'_B. Let L' be the total length of the two paths after this modification. We can then note the following about the reduction in length from this modification: $L - L' = (|a_1^* b_1^*| + |b_2^* a_2^*| + |b_1^\dagger b_2^\dagger|) - (|a_1^* a_2^*| + |b_1^\dagger b_1^*| + |b_2^* b_2^\dagger|)$.

Define $\omega = \delta(A, B)$. Using the definitions of well separation and rules of a metric space we

have:

$$|a_1^* b_1^*| \geq \omega, \quad |b_2^* a_2^*| \geq \omega, \quad |a_1^* a_2^*| \leq \frac{\omega}{s}, \quad |b_1^\dagger b_1^*| \leq \frac{\omega}{s}, \quad |b_2^* b_2^\dagger| \leq \frac{\omega}{s}, \quad |b_1^\dagger b_2^\dagger| \geq 0,$$

We can then conclude the following:

$$\begin{aligned} L - L' &= (|a_1^* b_1^*| + |b_2^* a_2^*| + |b_1^\dagger b_2^\dagger|) - (|a_1^* a_2^*| + |b_1^\dagger b_1^*| + |b_2^* b_2^\dagger|) \\ &\geq (|a_1^* b_1^*| + |b_2^* a_2^*|) - (|a_1^* a_2^*| + |b_1^\dagger b_1^*| + |b_2^* b_2^\dagger|) \\ &\geq 2\omega - (|a_1^* a_2^*| + |b_1^\dagger b_1^*| + |b_2^* b_2^\dagger|) \\ &\geq 2\omega - \frac{3\omega}{s} \\ &= \omega\left(2 - \frac{3}{s}\right) \end{aligned}$$

Thus if $s > \frac{3}{2}$ then we have that $L - L' > 0$ which is a contradiction since we assumed that the original paths were optimal (ie $L - L' \leq 0$). If $s = \frac{3}{2}$ then we have that $L - L' \geq 0$ and since by optimality we have that $L - L' \leq 0$ we can conclude that $L - L' = 0$ and therefore the newly constructed paths are also an optimal solution. Since this new paths removed the only two edges connecting A to B we can conclude that there exists an optimal solution where neither path has an edge connecting A to B.

Combining case 1 and case 2 we have that if $s \geq \frac{3}{2}$ then there exists an optimal solution where neither path has an edge connecting A to B and that if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property. Furthermore it follows from case 1 that if $s \geq 1$ then there exists an optimal solution where there are at most two edges connecting A to B and if $s > 1$ then all optimal solutions must satisfy this property. □

3.2.2 Extension to k Paths of Fixed Endpoint Pairings

We can extend these results to any arbitrary number of paths whose endpoints are fixed beforehand for each path. To do so we utilize the previous lemmas. If all endpoints are in the same set, then we can apply Lemma 3.2.1 to show that exactly one path crosses and hits all of the other set and all other paths stay within the same set. If the endpoints are split between the two sets, then we can apply Lemmas 3.2.2 and 3.2.4 to show that a path crosses the two sets if and only if it has endpoints in different sets. Then, we can apply Theorem 3.1.1 to show that if a path has endpoints in different sets, then it must cross the two sets exactly once. We also extend our result to the case of multiple tours which have fixed endpoints.

Theorem 3.2.5. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq \frac{3}{2}$. Then the multiple optimal Hamiltonian paths problem of $k \geq 2$ paths where each paths endpoints are fixed and all endpoints are in A has a solution where a single path visits all points in B consecutively and all other paths only have points in A. Furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.*

Proof. Suppose that for some optimal solution there are at least two paths that have edges connecting A to B. Then we can define two new sets of points A', B' where A' is the set of points in A that are visited by these two paths and B' is the set of points in B that are visited by these two paths. Then we can note that A' and B' are still s-well-separated since they are subsets of A and B respectively. We can then define the endpoints of the first path as $a', a'' \in A$ and the endpoints of the second path as $a''', a'''' \in A$. If $s > \frac{3}{2}$ then this would contradict Lemma 3.2.1 since it states that if $s > \frac{3}{2}$ then all optimal solutions must have only a single path with edges connecting A to B. If $s = \frac{3}{2}$ then by Lemma 3.2.1 there exists an optimal solution to the subproblem of the points in

these two paths where only a single path has two edges connecting A to B and the other path has no edges connecting A to B. We can then repeatedly apply this logic to keep generating optimal solutions with fewer paths connecting A to B until we have an optimal solution where only a single path has edges connecting A to B and all other paths have no edges connecting A to B.

Finally we can apply Theorem 3.1.2 to the path that has edges connecting A to B to construct a solution where this path visits all points in B consecutively and furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property. □

Corollary 3.2.5.1. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq \frac{3}{2}$. Then the multiple optimal travelling salesman problem of $k \geq 2$ tours which has all depot nodes in A has a optimal solution where all tours but one stay within A and the remaining tour visits all points in B consecutively. Furthermore, if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.*

Theorem 3.2.6. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq \frac{3}{2}$. Then the multiple optimal Hamiltonian paths problem of $k \geq 2$ paths where each paths endpoints are fixed and at least one path has an endpoint in A and at least one path has an endpoint in B has a solution where paths which have both an endpoint in A and an endpoint in B have a single edge connecting A to B and all other paths only have points in A or only have points in B (if they start/end in A resp B). Furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.*

Proof. Assume that for some optimal solution there exists a path whose endpoints are just in A xor B and that path has an edge connecting A to B. WLOG define the endpoints of the path to be $a', a'' \in A$ (otherwise swap the names for A and B). Choose another path which has at least one

endpoint in B. We can then define A' to be the set of points in A that are visited by these two paths and B' to be the set of points in B that are visited by these two paths. Then we can note that A' and B' are still s-well-separated by Lemma 2.3.5.

Case 1: This path has both endpoints in B. Then we can apply Lemma 3.2.4 to construct a solution where neither path has an edge connecting A to B and furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.

Case 2: This path has one endpoint in A and one endpoint in B. Then we can apply Lemma 3.2.2 to construct a solution where only the second path has edges connecting A to B and furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.

By repeatedly applying cases 1 and 2 we can construct a solution where paths that have both endpoints in A xor B never cross from A to B. Next take some path which has both one endpoint $a' \in A$ and one endpoint $b' \in B$ (if it exists). Define A' to be the set of points in A that are visited by this path and B' to be the set of points in B that are visited by this path. Then we can note that A' and B' are still s-well-separated by Lemma 2.3.5. It then follows from Theorem 3.1.1 that the optimal solution must have that this path only has a single edge connecting A to B.

Therefore we have that if $s \geq \frac{3}{2}$ then there exists an optimal solution where paths which have both an endpoint in A and an endpoint in B have a single edge connecting A to B and all other paths only have points in A or only have points in B (if they start/end in A resp B) and if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property. □

Corollary 3.2.6.1. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq \frac{3}{2}$. Then the multiple optimal travelling salesman problem of $k \geq 2$ tours which has a mixture of depot nodes in A and B has a solution where there are no edges connecting A to B in any of*

the tours. Furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.

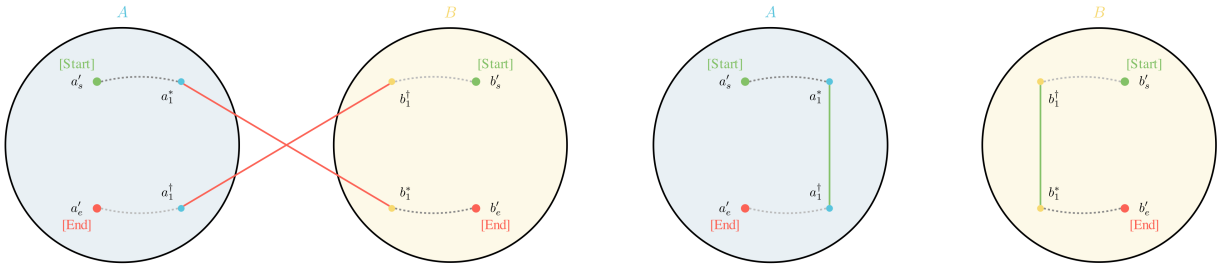
3.3 Multi Paths of Fixed Start and End Sets

To begin extending our results to other areas of interest, we will first observe the behavior of multiple paths problems where two sets of points—one of starting and one of ending—must map to each other via disjoint paths over two well-separated sets of points.

3.3.1 Two Entrances and Two Exits

To begin working towards a more general theorem, we will first show that if we have two well-separated sets of points and a pair of starting points and a pair of ending points, the optimal paths must have a specific structure depending on which sets the points are in. In this case we can control which path is connected to which set of endpoints. Therefore our first goal during each proof will be to show that optimal solution has a given endpoint configuration. An illustration of how to obtain a specific configuration of endpoints is shown in fig. 3.5 where a 2-opt swap is conducted across paths so that one path has both endpoints in A and the other path has both endpoints in B. Then once this endpoint configuration is established we can apply the techniques from the previous sections, namely Lemmas 3.2.1 to 3.2.4 and Theorems 3.1.1 and 3.1.2.

Lemma 3.3.1. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$. Then the multiple optimal Hamiltonian paths problem solution between starting nodes $a'_s, a''_s \in A$ and ending nodes $a'_e, a''_e \in A$ has a solution where one path only has points in A and the other visits all points in B consecutively. Furthermore if $s > 1$ then all optimal solutions must satisfy this property.*



(a) Each path starts in a different set and then crosses into and ends in the opposite set.

(b) One path starts and ends in A and the other path starts and ends in B.

Figure 3.5: By performing a 2-opt swap across the paths we can move from the configuration in (a) to the configuration in (b). The WSP condition ensures that the new connections are shorter than the old connections, leading to a shorter overall solution.

Proof. Assume that for some solution to the problem the two optimal paths each visit some points in B. We will define the set of points in A, B from the first path as A^*, B^* and the set of points in A, B from the second path as $A^\dagger, B^\dagger$. Then we can note that there must be the following edges in the first path: $(a_1^*, b_1^*), (b_2^*, a_2^*)$ and the following edges in the other path: $(a_1^\dagger, b_1^\dagger), (b_2^\dagger, a_2^\dagger)$ all ordered as they appear in their respective paths. Let the first path have endpoints a_s' and a_e' and let the second path have endpoints a_s'' and a_e''. This gives the following partial structure of the paths:

$$a_s' \rightsquigarrow a_1^* \to b_1^* \rightsquigarrow b_2^* \to a_2^* \rightsquigarrow a_e' \qquad a_s'' \rightsquigarrow a_1^\dagger \to b_1^\dagger \rightsquigarrow b_2^\dagger \to a_2^\dagger \rightsquigarrow a_e''$$

Since the paths are optimal we know that they satisfy the 2-optimality condition implying the following: $|a_1^* b_1^*| + |b_2^\dagger a_2^\dagger| \leq |a_1^* a_2^\dagger| + |b_2^\dagger b_1^*|$. This is to say that the following paths must have a combined length at least as long as the original paths:

$$a_s' \rightsquigarrow a_1^* \to a_2^\dagger \rightsquigarrow a_e'' \qquad a_s'' \rightsquigarrow a_1^\dagger \to b_1^\dagger \rightsquigarrow b_2^\dagger \to b_1^* \rightsquigarrow b_2^* \to a_2^* \rightsquigarrow a_e'$$

Since A and B are s-well-separated we know that $\forall i, j, k; |a_i a_j| \leq \frac{1}{s}|a_i b_k|$ (and vice versa for B). From this we can conclude the following:

$$\begin{aligned} |a_1^* b_1^*| + |b_2^\dagger a_2^\dagger| &\leq |a_1^* a_2^\dagger| + |b_2^\dagger b_1^*| \\ &\leq \frac{1}{s}|a_1^* b_1^*| + \frac{1}{s}|b_2^\dagger a_2^\dagger| \\ &= \frac{1}{s}(|a_1^* b_1^*| + |b_2^\dagger a_2^\dagger|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s > 1$ this is a contradiction and therefore any solution where both paths visit some points in B cannot be optimal. If $s = 1$ then we can conclude that $|a_1^* b_1^*| + |b_2^\dagger a_2^\dagger| = |a_1^* a_2^\dagger| + |b_2^\dagger b_1^*|$ which implies that the newly constructed paths are also optimal. Since the total number of loops between A and B has strictly decreased we can apply the same argument to construct an optimal solution where one path only has points in A and the other visits all points in B.

From here we can examine the path that visits all the points in B and take all the points it visits in A and label this as A'. We can then note that A' and B are s-well-separated by Lemma 2.3.5. By Theorem 3.1.2 we know that there exists an optimal Hamiltonian path between the two endpoints in A' that visits all points in B consecutively and furthermore if $s > 1$ then all optimal Hamiltonian paths must visit all points in B consecutively. □

Lemma 3.3.2. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$. Then the multiple optimal Hamiltonian paths problem solution between starting nodes $a'_s, a''_s \in A$ and ending nodes $a'_e \in A$ and $b'_e \in B$ has a solution where one path only has points in A and the other has exactly one edge connecting A to B. Furthermore if $s > 1$ then all optimal solutions must satisfy this property.*

Proof. Assume that for some solution to the problem the two optimal paths each visit some points in B. We will define the set of points in A, B from the first path as A^*, B^* and the set of points in A, B from the second path as $A^\dagger, B^\dagger$. Then we can note that there must be the following edges in the first path: $(a_1^*, b_1^*), (b_2^*, a_2^*)$ and the following edges in the other path: $(a_1^\dagger, b_1^\dagger)$ all ordered as they appear in their respective paths. WLOG let the first path have endpoints a_s' and a_e' and let the second path have endpoints a_s'' and b_e'. This gives the following partial structure of the paths:

$$a_s' \rightsquigarrow a_1^* \to b_1^* \rightsquigarrow b_2^* \to a_2^* \rightsquigarrow a_e' \qquad a_s'' \rightsquigarrow a_1^\dagger \to b_1^\dagger \rightsquigarrow b_e'$$

Since the paths are optimal we know that they satisfy the 2-optimality condition implying the following: $|b_2^* a_2^*| + |a_1^\dagger b_1^\dagger| \leq |b_2^* b_1^\dagger| + |a_1^\dagger a_2^*|$. This is to say that the following paths must have a combined length at least as long as the original paths:

$$a_s' \rightsquigarrow a_1^* \to b_1^* \rightsquigarrow b_2^* \to b_1^\dagger \rightsquigarrow b_e' \qquad a_s'' \rightsquigarrow a_1^\dagger \to a_2^* \rightsquigarrow a_e'$$

Since A and B are s-well-separated we know that $\forall i, j, k; |a_i a_j| \leq \frac{1}{s}|a_i b_k|$ (and vice versa for B). From this we can conclude the following:

$$\begin{aligned} |b_2^* a_2^*| + |a_1^\dagger b_1^\dagger| &\leq |b_2^* b_1^\dagger| + |a_1^\dagger a_2^*| \\ &\leq \frac{1}{s}|b_2^* a_2^*| + \frac{1}{s}|a_1^\dagger b_1^\dagger| \\ &= \frac{1}{s}(|b_2^* a_2^*| + |a_1^\dagger b_1^\dagger|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s > 1$ this is a contradiction and

therefore any solution where both paths visit some points in B cannot be optimal. If $s = 1$ then we can conclude that $|b_2^* a_2^*| + |a_1^\dagger b_1^\dagger| = |b_2^* b_1^\dagger| + |a_1^\dagger a_2^*|$ which implies that the newly constructed paths are also optimal. It follows via induction that all the extra loops between A and B can be removed from one of the paths to construct a solution where one path only has points in A and the other visits all points in B.

From here we can assume that we have a solution where one path has only points in A and the other path visits all the points in B. We can then note that A' and B are s-well-separated by Lemma 2.3.5. By Theorem 3.1.1 we know that there exists an optimal Hamiltonian path between the two endpoints in $A' \cup B$ that visits all points in A' before any point in B and furthermore if $s > 1$ then all optimal Hamiltonian paths must visit all points in A' before any point in B. □

Lemma 3.3.3. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$. Then the multiple optimal Hamiltonian paths problem solution between starting nodes $a'_s, a''_s \in A$ and ending nodes $b'_e, b''_e \in B$ has a solution where each path has exactly one edge connecting A to B. Furthermore if $s > 1$ then all optimal solutions must satisfy this property.*

Proof. Suppose that for some optimal solution one of its paths has more than one edge connecting A to B. Then we can define the set of points in A, B from this path as A^*, B^*. Then we can note that A^* and B^* are still s-well-separated by Lemma 2.3.5. If $s > 1$ then such a solution would contradict Theorem 3.1.1 as applied to A^* and B^*. If $s = 1$ then we can apply Theorem 3.1.1 to A^* and B^* to construct a new path over $A^* \cup B^*$ that only has a single edge connecting A^* to B^* and is also optimal. We can then swap this new path with the original path in the previous solution to construct where the path that previously had more than one edge connecting A to B now only has a single edge connecting A to B.

Observing these results across both paths we can conclude that there exists an optimal solution where both paths only have a single edge connecting A to B and that if $s > 1$ then all optimal solutions must satisfy this property. □

The following theorem demonstrates that if the start and end points are distributed such that each set (A and B) contains one start and one end point, the optimal paths never cross between the sets. This proof departs from the techniques used in the previous proofs. Specifically, while preceding results typically required $s \geq 1$ this theorem requires $s \geq \frac{3}{2}$ to ensure the desired optimal structure; notably, this structure remains invariant as s increases beyond $\frac{3}{2}$. If s were only greater than 1 then it could only be guarantted that paths must start and end in the same set and at most one path could wander through the other set. To rule out this "wandering," we show that such a subpath could be spliced and inserted to the other path to produce a lower-cost solution.

Lemma 3.3.4. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq \frac{3}{2}$. Then the optimal multiple Hamiltonian paths problem solution between starting nodes $a'_s \in A$ and $b'_s \in B$ and ending nodes $a'_e \in A$ and $b'_e \in B$ has a solution where neither path has an edge connecting A to B. Furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property. Furthermore no optimal solution can have both paths connecting A and B.*

Proof. Assume that for some optimal solution one of its path has an edge connecting A to B.

Case 1: The path has one endpoint in A and the other in B. Let the first path have endpoints a'_s and b'_e and let the second path have endpoints b'_s and a'_e. Then we can note that there must be the following edges in the path: (a_1^*, b_1^*) in the first path and in the other path $(b_1^\dagger, a_1^\dagger)$ all ordered as

they appear in their respective paths. This gives the following partial structure of the paths:

$$a'_s \rightsquigarrow a^*_1 \rightarrow b^*_1 \rightsquigarrow b'_e \qquad b'_s \rightsquigarrow b^\dagger_1 \rightarrow a^\dagger_1 \rightsquigarrow a'_e$$

Since the paths are optimal we know that they satisfy the 2-optimality condition implying the following: $|a^*_1 b^*_1| + |b^\dagger_1 a^\dagger_1| \leq |a^*_1 a^\dagger_1| + |b^\dagger_1 b^*_1|$. This is to say that the following paths must have a combined length at least as long as the original paths:

$$a'_s \rightsquigarrow a^*_1 \rightarrow a^\dagger_1 \rightsquigarrow a'_e \qquad b'_s \rightsquigarrow b^\dagger_1 \rightarrow b^*_1 \rightsquigarrow b'_e$$

Since A and B are s-well-separated we know that $\forall i, j, k; |a_i a_j| \leq \frac{1}{s}|a_i b_k|$ (and vice versa for B). From this we can conclude the following:

$$\begin{aligned} |a^*_1 b^*_1| + |b^\dagger_1 a^\dagger_1| &\leq |a^*_1 a^\dagger_1| + |b^\dagger_1 b^*_1| \\ &\leq \frac{1}{s}|a^*_1 b^*_1| + \frac{1}{s}|b^\dagger_1 a^\dagger_1| \\ &= \frac{1}{s}(|a^*_1 b^*_1| + |b^\dagger_1 a^\dagger_1|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. This however is a contradiction since we assumed that $s \geq \frac{3}{2}$.

Case 2: The path has both endpoints in either A xor B. WLOG assume that the path which starts at a'_s and ends at a'_e and traverses part of B. We will call this path P_A and the other path P_B.

It then follows from Lemma 3.2.4 that if $s \geq \frac{3}{2}$ then there exists an optimal solution where neither path has an edge connecting A to B and that if $s > \frac{3}{2}$ then all optimal solutions must satisfy

this property.

Combining case 1 and case 2 we have that if $s \geq \frac{3}{2}$ then there exists an optimal solution where neither path has an edge connecting A to B and that if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property. Furthermore it follows from case 1 that there exists an optimal solution where the endpoints of each path are in the same set if $s \geq 1$ and that if $s > 1$ then all optimal solutions must satisfy this property. □

To show the required bound is tight, we can craft a set of five points in the 2D Euclidean plane such that the two sets are s-well-separated but the optimal solution to the multiple optimal paths problem would require one path to wander away from its starting/ending set to the other set and then back to its starting/ending set.

Remark. *The bound on s in Lemma 3.3.4 is tight.*

Proof. Let $s < \frac{3}{2}$. Let $a_1 = (0,0), a_2 = \left(0, \frac{2}{3}\right)$ and $b_1 = \left(\frac{2\sqrt{2}}{3} + \frac{1}{s}, \frac{1}{3}\right), b_2 = \left(\frac{2\sqrt{2}}{3}, \frac{1}{3}\right), b_3 = \left(\frac{2\sqrt{2}}{3} + \frac{1}{s}, \frac{1}{3}\right)$. Next we define $A = \{a_1, a_2\}$ and $B = \{b_1, b_2, b_3\}$. We can then note that the diameter of A is $\frac{2}{3}$ and the diameter of B is $\frac{1}{s}$. The distance between the two sets is 1 since $|a_1 b_2| = |a_2 b_2| = 1$ and all other distances between points in A and B are strictly greater than 1. Thus the separation factor of these two sets is $\frac{1}{\frac{1}{s}} = s$.

Consider the multiple optimal Hamiltonian paths problem between starting nodes a_1, b_1 and ending nodes a_2, b_3. We can then compare the following two solutions to this problem (which represent the only two viable solutions):

- $P_A : a_1 \to a_2$ and $P_B : b_1 \to b_2 \to b_3$
- $P'_A : a_1 \to b_2 \to a_2$ and $P'_B : b_1 \to b_3$

We can then note that the total length of the second solution is $1 + 1 = 2$. The total length of the first solution is $\frac{2}{3} + \frac{1}{s} + \frac{1}{s} = \frac{2}{3} + \frac{2}{s} > 2$ since $s < \frac{3}{2}$. Therefore the first solution is not optimal and thus there does not exist an optimal solution where neither path has an edge connecting A to B. □

3.3.2 Extension to k Paths of Fixed Start and End Sets

Next, we can utilize the previous lemmas to show that if we have k paths with fixed starting and ending sets then the optimal solution must have a specific structure. Specifically, if all starts and ends are in the same set then all but one path must stay within that set, while the remaining path must visit all points in the other set consecutively. If the starts and ends are split between the two sets then the optimal solution will maximize the number of paths that can stay within a single set and then require the remaining paths to cross between the two sets exactly once.

Theorem 3.3.5. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$. Then the multiple optimal Hamiltonian paths problem of $k \geq 2$ paths which has all starting and ending nodes in A has a optimal solution where all paths but one stay within A and the remaining path visits all points in B consecutively. Furthermore if $s > 1$ then all optimal solutions must satisfy this property.*

Proof. Suppose that for some optimal solution there are at least two paths that visit some points in B. Then we can define two new sets of points A', B' where A' is the set of points in A that are visited by these two paths and B' is the set of points in B that are visited by these two paths. Then we can note that A' and B' are still s-well-separated by Lemma 2.3.5. If $s > 1$ then this would cause a contradiction as such a solution would also be optimal on the subproblem and thus violate

Lemma 3.3.1. If $s = 1$ then we can apply Lemma 3.3.1 to construct a solution to the subproblem where one path visits only points in A' and the other path visits all points in B' consecutively.

We can then apply this logic repeatedly between the current multi set visiting path and any other non just A visiting path to construct a solution where a single path visits all points in B and all other paths only have points in A. We can then apply Theorem 3.1.2 to the path that visits some points in B to construct a solution where this path visits all points in B consecutively and furthermore if $s > 1$ then all optimal solutions must satisfy this property. □

Theorem 3.3.6. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq \frac{3}{2}$. Then the multiple optimal Hamiltonian paths problem of $k \geq 2$ paths which has n starting nodes in A and m ending nodes in A with $\min(n, m) \neq k$ and $0 \neq \max(n, m)$ has a solution where there are $|n - m|$ paths with exactly one edge connecting A to B. There are $\min(n, m)$ paths with only points in A and there are $k - \max(n, m)$ paths with only points in B. Furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.*

Proof. Say we take a given optimal solution to the problem.

Case 1: There are not $|n - m|$ paths with one endpoint in A and the other in B.

Case 1a: There are less than $|n - m|$ paths with one endpoint in A and the other in B. This is impossible since if there are less than $|n - m|$ paths with one endpoint in A and the other in B then there must be more than $\min(n, m)$ paths with only endpoints points in A or more than $k - \max(n, m)$ paths with only endpoints in B which is a contradiction since those values represent the maximum number of paths that can have only endpoints in A and B respectively.

Case 1b: There are more than $|n - m|$ paths with one endpoint in A and the other in B. In order for this extra path to come about there must be at least one less path from the $\min(n, m)$

paths with only points in A and one less path from the $k - \max(n, m)$ paths with only points in B. This would then imply that there are at least two extra paths with one endpoint in A and the other in B. However by Lemma 3.3.4 we know that if $s > \frac{3}{2}$ then there cannot be two paths with one endpoint in A and the other in B and if $s = \frac{3}{2}$ then there exists an optimal solution where these two extra paths are replaced by two paths that start and end in the same sets (one in A and one in B). It follows from induction that if $s = \frac{3}{2}$ then there exists an optimal solution where there are exactly $|n - m|$ paths with one endpoint in A and the other in B.

Case 2: There are less than $\min(n, m)$ paths with only endpoints points in A or less than $k - \max(n, m)$ paths with only endpoints in B. For this to be the case there must be at least one more path with one endpoint in A and the other in B than $|n - m|$ which is a contradiction since then case 1. would have applied.

Case 3: There are exactly $|n - m|$ paths with one endpoint in A and the other in B and there are exactly $\min(n, m)$ paths with only endpoints points in A and exactly $k - \max(n, m)$ paths with only endpoints in B.

Case 3a: There is a path with one endpoint in A and the other in B that has more than one edge connecting A to B. This is impossible since by Theorem 3.1.1 we know that such a path cannot be optimal when $s > 1$.

Case 3b: There is a path with both endpoints in A xor B that has an edge connecting A to B. WLOG assume that there is a path with both endpoints in A that has an edge connecting A to B. Since $\min(n, m) \neq k$ and $0 \neq \max(n, m)$ we know that there must exist at least one other path with an endpoint from B

Case 3b.i: $|m - n| > 0$. We can then choose a single path from the $|n - m|$ paths with one endpoint in A and the other in B and add it to the path with both endpoints in A that has an edge

connecting A to B to construct a new multiple Hamiltonian paths problem of the points from these two paths. It then follows from Lemma 3.3.2 that if the optimal solution to this new problem had the original path with both endpoints in A travelling to B this would be a contradiction since such a path cannot be optimal when $s > 1$.

Case 3b.ii: $|m - n| = 0$. Then since another path must exist this path must have both endpoints in B. We can then form a new multiple Hamiltonian paths problem of the points from these two paths. It then follows from Lemma 3.3.4 that if $s > \frac{3}{2}$ then having the original path cross into B is impossible. If $s = \frac{3}{2}$ then there exists an optimal solution to this new problem where neither path has an edge connecting A to B. It then follows by induction that for all paths with the endpoints in the same set (either A or B) there exists an optimal solution where none of these paths have an edge connecting A to B.

If $s = \frac{3}{2}$ then there exists an optimal solution that satisfies the property that there are $|n-m|$ paths with exactly one edge connecting A to B, there are $\min(n, m)$ paths with only points in A and there are $k - \max(n, m)$ paths with only points in B. If $s > \frac{3}{2}$ then all optimal solutions must satisfy the previous properties. □

3.4 Multi Paths of Fixed Endpoint Sets

The natural extension of the previous section is to relax the requirement that the multiple paths all start from one fixed set and map to another. Instead, we only require that the endpoints of a given path are drawn from a predefined common endpoint set such that every point from the endpoint set is used once. More formally we have k paths and $2k$ endpoints such that each endpoint must belong to some path. In this relaxation we show that the optimal solution is able to

more aggressively eliminate crossing between the two sets.

3.4.1 Two Paths / Four Endpoints

We begin by demonstrating that for two paths and four endpoints, the optimal solution follows a specific structure. We analyze three distinct cases: (1) all endpoints reside in the same set, (2) three endpoints are in one set and one endpoint is in the other set, and (3) each set contains two endpoints. The proofs required for these cases primarily show that any deviation from the prescribed structure would contradict the lemmas established in the previous section, specifically Lemmas 3.3.1, 3.3.2 and 3.3.4.

Lemma 3.4.1. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$. Then the multiple optimal Hamiltonian paths problem solution between endpoints $a', a'', a''', a'''' \in A$ has a solution where only a single path has two edges connecting A to B and the other path has no edges connecting A to B. Furthermore if $s > 1$ then all optimal solutions must satisfy this property.*

Proof. Consider two labellings for the endpoints the first where a', a'' are required starting nodes and a''', a'''' are required ending nodes and the second where a', a''' are required starting nodes and a'', a'''' are required ending nodes.

Then any optimal solution must fall into one of the previous cases. It follows from Lemma 3.3.1 that under either case if $s > 1$ then any optimal solution must have exactly two edges connecting A to B and if $s = 1$ then there exists an optimal solution where there are exactly two edges connecting A to B. □

Lemma 3.4.2. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$.*

Then the multiple optimal Hamiltonian paths problem solution between endpoints $a', a'', a''' \in A$ and $b' \in B$ has a solution where one path only has points in A and the other has exactly one edge connecting A to B. Furthermore if $s > 1$ then all optimal solutions must satisfy this property.

Proof. WLOG assume that for some optimal solution to the problem we have one path with endpoints a', a'' and the other path with endpoints a''', b' (if this is not the case then we can just shuffle the a endpoint names). If $s = 1$ and the solution doesn't have a path that visits all of B consecutively then we can apply Lemma 3.3.2 to construct a optimal solution where one path only has points in A and the other path visits all points in B consecutively. If $s > 1$ and the solution doesn't have a path that visits all of B consecutively then this would be a contradiction since by Lemma 3.3.2 we know that generating a optimal path with starting nodes a', a''' and ending nodes a'', b' that has more than one edge connecting A to B is impossible. □

The proof of the following case follows a different approach: we first demonstrate that in an optimal configuration, each path must have both endpoints within the same set. Once this property is established, the final result follows directly from Lemma 3.3.4.

Lemma 3.4.3. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq \frac{3}{2}$. Then the multiple optimal Hamiltonian paths problem solution over endpoints $a', a'' \in A$ and $b', b'' \in B$ has a solution where there are no edges connecting A to B. Furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.*

Proof. Assume that for some optimal solution it has an edge connecting A to B.

Case 1: The path with an edge connecting A to B has endpoints in different sets. Then the other path must also have endpoints in different sets. WLOG let one path have endpoints a', b' and the other have endpoints a'', b'' (if not then we can shuffle the endpoint names). Then we can note

that there must exist one edge of the form (a^*, b^*) in the first path and one edge of the form $(a^\dagger, b^\dagger)$ in the second path where $a^*, a^\dagger \in A$ and $b^*, b^\dagger \in B$. This gives the following partial structure to the paths:

$$a' \rightsquigarrow a^* \to b^* \rightsquigarrow b' \qquad a'' \rightsquigarrow a^\dagger \to b^\dagger \rightsquigarrow b''$$

Since the paths are optimal we know that they satisfy the 2-optimality condition implying that $|a^*b^*| + |a^\dagger b^\dagger| \leq |a^*a^\dagger| + |b^*b^\dagger|$. That is to say the following paths must have a combined length at least as long as the original paths:

$$a' \rightsquigarrow a^* \to a^\dagger \rightsquigarrow a'' \qquad b' \rightsquigarrow b^* \to b^\dagger \rightsquigarrow b''$$

Since A and B are s-well-separated we know that $\forall i, j, k; |a_i a_j| \leq \frac{1}{s}|a_i b_k|$ (and vice versa for B). From this we can conclude the following:

$$\begin{aligned} |a^*b^*| + |a^\dagger b^\dagger| &\leq |a^*a^\dagger| + |b^*b^\dagger| \\ &\leq \frac{1}{s}|a^*b^*| + \frac{1}{s}|a^\dagger b^\dagger| \\ &= \frac{1}{s}(|a^*b^*| + |a^\dagger b^\dagger|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s > 1$ this is a contradiction and therefore any solution where a path has endpoints in different sets cannot be optimal. If $s = 1$ then we can conclude that $|a^*b^*| + |a^\dagger b^\dagger| = |a^*a^\dagger| + |b^*b^\dagger|$ which implies that the newly constructed paths are also optimal.

Case 2: The path with an edge connecting A to B has endpoints in the same set. Then the other path must also have endpoints in the same set. WLOG let one path have endpoints a', a'' and

the other have endpoints b', b'' (if not then we can shuffle the endpoint names). We can then label a', b' as the starting nodes and a'', b'' as the ending nodes. It then follows from Lemma 3.3.4 that we can construct a solution where there are no edges connecting A to B and furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.

If $s > \frac{3}{2}$ then either case leads to a contradiction and thus all optimal solutions must have no edges connecting A to B. If $s = \frac{3}{2}$ then either case leads to a optimal solution where there are no edges connecting A to B. □

3.4.2 Extension to k Paths / $2k$ Endpoints

We can extend the previous results to any arbitrary number of paths k whose endpoints are drawn without replacement from a set of $2k$. To do so, we utilize the previous lemmas. In this case, we can see that the problems will fall into one of three categories: first, when all endpoints are in the same set; second, when there is an odd number of endpoints in each set; and third, when there is an even number of endpoints in each set. These cases result in two edges, one edge, and no edges between the two sets, respectively.

Theorem 3.4.4. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$ and we have a multiple optimal Hamiltonian paths problem with $k \geq 2$ paths where all endpoints are in A. Then there exists an optimal solution where a single path visits all points in B consecutively and all other paths only have points in A. Furthermore if $s > 1$ then all optimal solutions must satisfy this property.*

Proof. Consider some optimal solution to the problem. Suppose that there are at least two paths that have edges connecting A to B. Then we can define two new sets of points A', B' where A'

is the set of points in A that are visited by these two paths and B' is the set of points in B that are visited by these two paths. Then we can note that A' and B' are still s-well-separated since they are subsets of A and B respectively. By Lemma 3.4.1 we know that there exists an optimal solution to the subproblem of the points in these two paths where only a single path has two edges connecting A to B and the other path has no edges connecting A to B and furthermore if $s > 1$ then all optimal solutions to this subproblem must satisfy this property. We can then apply this logic repeatedly to construct a solution where only a single path has edges connecting A to B and all other paths have no edges connecting A to B. We can then apply Theorem 3.1.2 to the path that has edges connecting A to B to construct a solution where this path visits all points in B consecutively and furthermore if $s > 1$ then all optimal solutions must satisfy this property. □

Corollary 3.4.4.1. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$. Then the multiple optimal travelling salesman problem of $k \geq 2$ tours which has all depots out of a single node has a optimal solution where all tours but one stay within A and the remaining tour visits all points in B consecutively. Furthermore, if $s > 1$ then all optimal tours must satisfy this property.*

Theorem 3.4.5. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq 1$ and we have a multiple optimal Hamiltonian paths problem with $k \geq 2$ paths where there are an odd number of endpoints in A. Then there exists an optimal solution where a single path has exactly one edge connecting A to B and all other paths have no edges connecting A to B. Furthermore if $s > 1$ then all optimal solutions must satisfy this property.*

Proof. Consider some optimal solution to the problem. Suppose that there are at least two paths that have endpoints in different sets (one in A and the other in B). Suppose that the first path has

endpoints a', b' and the second path has endpoints a'', b''. Then we can note that there must exist one edge of the form (a^*, b^*) in the first path and one edge of the form $(a^\dagger, b^\dagger)$ in the second path where $a^*, a^\dagger \in A$ and $b^*, b^\dagger \in B$. This path has the following partial structure:

$$a' \rightsquigarrow a^* \rightarrow b^* \rightsquigarrow b' \qquad a'' \rightsquigarrow a^\dagger \rightarrow b^\dagger \rightsquigarrow b''$$

Since the paths are optimal we know that they satisfy the 2-optimality condition implying the following:

$$|a^*b^*| + |a^\dagger b^\dagger| \leq |a^*a^\dagger| + |b^*b^\dagger|$$

This is to say that the following paths must have a combined length at least as long as the original paths:

$$a' \rightsquigarrow a^* \rightarrow a^\dagger \rightsquigarrow a'' \qquad b' \rightsquigarrow b^* \rightarrow b^\dagger \rightsquigarrow b''$$

Since A and B are s-well-separated we know that $\forall i, j, k; |a_i a_j| \leq \frac{1}{s}|a_i b_k|$ (and vice versa for B). From this we can conclude the following:

$$\begin{aligned} |a^*b^*| + |a^\dagger b^\dagger| &\leq |a^*a^\dagger| + |b^*b^\dagger| \\ &\leq \frac{1}{s}|a^*b^*| + \frac{1}{s}|a^\dagger b^\dagger| \\ &= \frac{1}{s}(|a^*b^*| + |a^\dagger b^\dagger|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s > 1$ this is a contradiction and therefore any solution where there are at least two paths with endpoints in different sets cannot be optimal. If $s = 1$ then we can conclude that $|a^*b^*| + |a^\dagger b^\dagger| = |a^*a^\dagger| + |b^*b^\dagger|$ which implies

that the newly constructed paths are also optimal. It follows that all but one path with endpoints in different sets can be replaced by paths with endpoints in the same set and thus there exists an optimal solution where only a single path has endpoints in different sets.

We can then take this path with endpoints in different sets and any other path with endpoints in the same set and create two new sets of points A', B' where A' is the set of points in A that are visited by these two paths and B' is the set of points in B that are visited by these two paths. Then we can note that A' and B' are still s-well-separated since they are subsets of A and B respectively. We can then apply Lemma 3.4.2 to construct a solution to the subproblem of the points in these two paths where one has a single edge connecting A to B and the other has no edges connecting A to B and furthermore if $s > 1$ then all optimal solutions to this subproblem must satisfy this property. We can then apply this logic repeatedly to construct a solution where only a single path has an edge connecting A to B and all other paths have no edges connecting A to B. □

Theorem 3.4.6. *Suppose we have two sets of points A and B that are s-well-separated where $s \geq \frac{3}{2}$ and we have a multiple optimal Hamiltonian paths problem with $k \geq 2$ paths where there are a non-zero even number of endpoints in A and a non-zero even number of endpoints in B. Then there exists an optimal solution where there are no edges connecting A to B in any of the paths. Furthermore if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.*

Proof. Consider some optimal solution to the problem. Suppose that there are at least two paths that have endpoints in different sets (one in A and the other in B). Let the first path have endpoints a', b' and the second path have endpoints a'', b''. Next we can define A', B' where A' is the set of points in A that are visited by these two paths and B' is the set of points in B that are visited by these two paths. Then we can note that A' and B' are still s-well-separated since they are subsets

of A and B respectively. We can then apply Lemma 3.4.3 to construct a solution to the subproblem of the points in these two paths where each path has its endpoints in the same set furthermore if $s > \frac{3}{2}$ then all optimal solutions to this subproblem must satisfy this property. We can then apply this logic repeatedly to construct a solution where there are no paths with endpoints in different sets.

Then for any path with endpoints both in A (resp. B) we can take any other path with endpoints both in B (resp. A) and apply Lemma 3.4.3 to construct a solution to the subproblem of the points in these two paths where neither path has an edge connecting A to B and furthermore if $s > \frac{3}{2}$ then all optimal solutions to this subproblem must satisfy this property. We can then apply this logic repeatedly to construct a solution where there are no edges connecting A to B in any of the paths. □

3.5 Multi Paths of Cycle Forming Predefined Endpoint Pairs

While the previous section maximally relaxes the requirements on the endpoints of the paths, we can also consider a more restrictive case where the endpoints are drawn from a predefined set of pairs, and these pairs must be arranged to form a single cycle. The objective is then to find the minimal total length of the connecting paths, subject to the constraint that alternating between the paths and the predefined pairs forms exactly one continuous cycle and all points are visited. This problem is of particular interest as it can be used to expose the behavior of a tour as it passes in and out of specific region which is made up of two well-separated sets of points (further illustrated in fig. 3.6). The previous setup from section 3.2 is ineffective as it allows for more direct crossings between the two sets than possible in the optimal tour. The setups from

sections 3.3 and 3.4 can result in optimal solutions that ultimately break apart the tour into multiple disconnected components (consider that a path going from $a_1 \to b_3$ in fig. 3.6 would be valid but also break the tour).

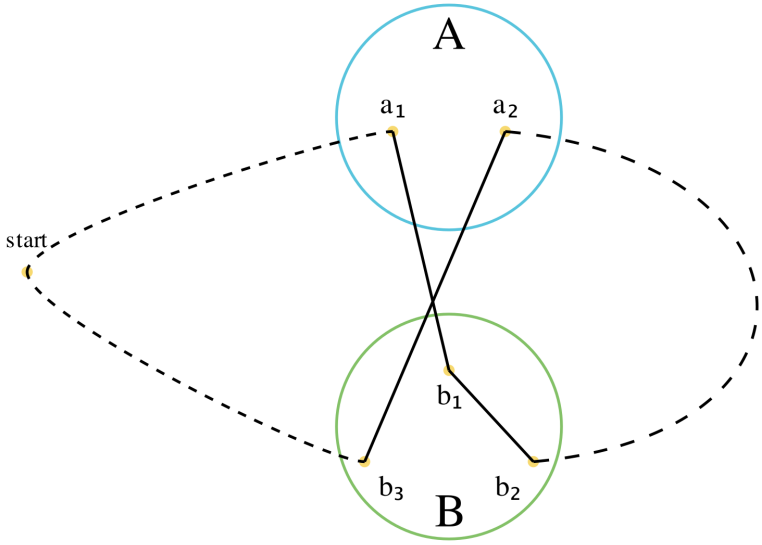


Figure 3.6: Behavior of a tour through two small well-separated sets. The dashed lines represent foreign parts of the tour outside of $A \cup B$ and solid lines represent the edges of the tour that are exclusively within $A \cup B$.

We demonstrate that problems will fall into one of three categories: first, when *no* predefined pair has endpoints in different sets; second, when there is a *odd* number of predefined pairs with endpoints in different sets; and third, when there is a *non-zero even* number of predefined pairs with endpoints in different sets. In these cases, there must exist an optimal solution with two, one, and zero edges between the two sets across the paths, respectively.

Theorem 3.5.1. *Suppose sets A and B are s-well-separated for $s \geq \frac{3}{2}$. Consider a multiple optimal Hamiltonian paths problem of $k \geq 2$ paths over $2k$ endpoints, organized into k pairs of endpoints. The objective is to minimize the total length of the k paths, subject to the constraint that alternating between the paths and the predefined pairs forms exactly one continuous cycle. Suppose that for each predefined pair of endpoints, the two endpoints are always in the same set*

(either A or B). Then there exists an optimal solution where across the entire solution only two edges cross between A and B. Furthermore, if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.

Proof. To prove this theorem we will first show that the optimal solution will place all the A endpoints consecutively in the cycle order and all the B endpoints consecutively in the cycle order. Then we will leverage the previous theorems to show that under this required ordering an optimal solution exists which only has two edges crossing between A and B.

To begin, consider some optimal solution to the problem.

Case 1: All endpoints are in the same set. Since the solution is optimal it is also an optimal solution to the multiple optimal Hamiltonian paths problem where the endpoints of each path are fixed as they are in the current solution. If $s > \frac{3}{2}$ then the current solution must have exactly two edges crossing between A and B by Theorem 3.2.5. If $s = \frac{3}{2}$ it follows that for the given configuration of endpoints there exists an optimal solution where only two edges cross between A and B by Theorem 3.2.5.

Case 2: There are some endpoints in A and some endpoints in B. Define the order of the endpoints in the cycle as $e_1, \ldots, e_{2k}$ and let $P_1, \ldots, P_k$ be the paths such that P_i connects e_{2i-1} to e_{2i}. Suppose that two endpoints e_j, e_{j+1} (where $e_{2k+1} = e_1$) connected via some path P_i in the solution are such that one is in A and the other is in B. All alternations between sets must happen via paths since the predefined pairs of endpoints are all in the same set. Furthermore, after completing the cycle we must return to the same set we started in, thus there must be an even number of such pairs. Since there is at least one predefined pair of endpoints in both sets there must be at least two paths where one endpoint is in A and the other is in B.

Assume by contradiction that there are strictly more than two paths in the solution where one endpoint is in A and the other is in B. Because the total number of such paths must be even, there must be at least 4 such paths. Let us consider any sequence of four such paths in the order they appear in the cycle. Because the predefined pairs never cross between sets, any sequence of crossing paths in cycle order must strictly alternate directions. Therefore, WLOG we can label the endpoints of these four paths as follows, the first path connects a_1 to b_1, the second path connects b_2 to a_2, the third path connects a_3 to b_3, and the fourth path connects b_4 to a_4 where $a_i \in A$ and $b_i \in B$ for all i.

Note that on the path that goes from a_1 to b_1 there must be an edge directly connecting from A to B and we will label this edge as $\{a^*, b^*\}$ and from a_3 to b_3 the same is true and we label this edge as $\{a^\dagger, b^\dagger\}$. Then we can note the cycle goes:

$$\cdots \sim \boxed{a_1 \rightsquigarrow a^* \to b^* \rightsquigarrow b_1} \sim \boxed{b_2 \rightsquigarrow a_2} \sim \boxed{a_3 \rightsquigarrow a^\dagger \to b^\dagger \rightsquigarrow b_3} \sim \boxed{b_4 \rightsquigarrow a_4} \sim \ldots.$$

It follows that we can break apart the solution at the two edges $\{a^*, b^*\}$ and $\{a^\dagger, b^\dagger\}$ and reverse the order of the cycle and paths along the cutout section between these two edges to get a new solution where the cycle goes:

$$\cdots \sim \boxed{a_1 \rightsquigarrow a^* \to a^\dagger \rightsquigarrow a_3} \sim \boxed{a_2 \rightsquigarrow b_2} \sim \boxed{b_1 \rightsquigarrow b^* \to b^\dagger \rightsquigarrow b_3} \sim \boxed{b_4 \rightsquigarrow a_4} \sim \ldots.$$

Because the predefined pairs are simply undirected pairings of endpoints within the exact same set, traversing them in reverse during this cycle modification is entirely valid and maintains all predefined pairings and does not create multiple cycles.

Since the previous solution was assumed optimal we know that $|a^*b^*| + |a^\dagger b^\dagger| \leq |a^*a^\dagger| + |b^*b^\dagger|$. Since A and B are s-well-separated we know that $\forall i, j, k; |a_i a_j| \leq \frac{1}{s}|a_i b_k|$ (and vice versa for B). From this we can conclude:

$$\begin{aligned} |a^*b^*| + |a^\dagger b^\dagger| &\leq |a^*a^\dagger| + |b^*b^\dagger| \\ &\leq \frac{1}{s}|a^*b^*| + \frac{1}{s}|a^\dagger b^\dagger| \\ &\leq \frac{1}{s}(|a^*b^*| + |a^\dagger b^\dagger|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which is a contradiction since $s \geq \frac{3}{2}$. Therefore there must be exactly two paths where one endpoint is in A and the other is in B as desired.

Since the solution is optimal it is also an optimal solution to the multiple optimal Hamiltonian paths problem where the endpoints of each path are fixed as they are in the current solution. If $s > \frac{3}{2}$ then the current solution must have exactly two edges crossing between A and B by Theorem 3.2.6. If $s = \frac{3}{2}$ it follows that for the given configuration of endpoints there exists an optimal solution where only two edges cross between A and B by Theorem 3.2.6.

Therefore in both cases we have shown that there exists an optimal solution where across the entire solution only two edges cross between A and B as desired and furthermore, if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property as desired. □

Theorem 3.5.2. *Suppose sets A and B are s-well-separated for $s \geq \frac{3}{2}$. Consider a multiple optimal Hamiltonian paths problem of $k \geq 2$ paths over $2k$ endpoints, organized into k pairs of endpoints. The objective is to minimize the total length of the k paths, subject to the constraint that alternating between the paths and the predefined pairs forms exactly one continuous cycle. Suppose that there is a **odd** number of predefined pairs with one point in A and the other in B.*

Then there exists an optimal solution where only a single edge crosses between A and B across the entire solution. Furthermore, if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.

Proof. To prove this we first show that in the optimal solution the endpoint cycle is ordered such that only a single path has endpoints in different sets. Then our previous theorems will imply that given the required ordering of the endpoints in the cycle, there exists an optimal solution where only a single edge crosses between A and B.

To begin, consider some optimal solution to the problem. Let us establish an arbitrary traversal direction along the cycle, against which the direction of all paths will be relative to. Since the total number of alternations between A and B along the cycle must be even and there is an odd number of predefined pairs with one point in A and the other in B, there must be an odd number of paths with endpoints in different sets. Assume by contradiction that there is more than one path in the solution that has endpoints in different sets. Thus there must be at least 3 such paths.

It follows via the pigeon hole principle that at least two of these paths must be in the same direction along the cycle. WLOG we can label these two paths as P_i and P_j where P_i connects $a' \in A$ to $b' \in B$ and P_j connects $a'' \in A$ to $b'' \in B$ and both paths are in the same direction along the cycle. We can then note that in both P_i and P_j there must be an edge directly connecting from A to B and we will label these edges as $\{a^*, b^*\}$ and $\{a^\dagger, b^\dagger\}$ respectively. Then we can note the cycle goes:

$$\cdots \sim \boxed{a' \rightsquigarrow a^* \to b^* \rightsquigarrow b'} \sim \cdots \sim \boxed{a'' \rightsquigarrow a^\dagger \to b^\dagger \rightsquigarrow b''} \sim \ldots.$$

It follows that we can break apart the solution at the two edges $\{a^*, b^*\}$ and $\{a^\dagger, b^\dagger\}$ and reverse the order of the cycle and paths along the cutout section between these two edges to get a new solution

where the cycle goes:

$$\cdots \sim \boxed{a' \rightsquigarrow a^* \rightarrow a^\dagger \rightsquigarrow a''} \sim \cdots \sim \boxed{b' \rightsquigarrow b^* \rightarrow b^\dagger \rightsquigarrow b''} \sim \ldots.$$

Because the predefined pairs are simply undirected pairings of endpoints, traversing them in reverse during this cycle modification is entirely valid and maintains all predefined pairs and does not create multiple cycles.

Since the previous solution was assumed optimal we know that $|a^*b^*| + |a^\dagger b^\dagger| \leq |a^*a^\dagger| + |b^*b^\dagger|$. Since A and B are s-well-separated we know that $\forall i, j, k; |a_i a_j| \leq \frac{1}{s}|a_i b_k|$ (and vice versa for B). From this we can conclude:

$$\begin{aligned} |a^*b^*| + |a^\dagger b^\dagger| &\leq |a^*a^\dagger| + |b^*b^\dagger| \\ &\leq \frac{1}{s}|a^*b^*| + \frac{1}{s}|a^\dagger b^\dagger| \\ &\leq \frac{1}{s}(|a^*b^*| + |a^\dagger b^\dagger|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which is a contradiction since $s \geq \frac{3}{2}$. Therefore there cannot be two paths in the same direction that have edges connecting A to B.

Since the solution is optimal it is also an optimal solution to the multiple optimal Hamiltonian paths problem where the endpoints of each path are fixed as they are in the current solution. If $s > \frac{3}{2}$ then there must only be a single edge crossing between A and B in the current solution by Theorem 3.2.6. If $s = \frac{3}{2}$ then there must exist an optimal solution which uses the same ordering of the endpoints as in the current solution and has only a single edge crossing between A and B by Theorem 3.2.6. □

Theorem 3.5.3. *Suppose sets A and B are s-well-separated for $s \geq \frac{3}{2}$. Consider a multiple optimal Hamiltonian paths problem of $k \geq 2$ paths over $2k$ endpoints, organized into k pairs of endpoints. The objective is to minimize the total length of the k paths, subject to the constraint that alternating between the paths and the predefined pairs forms exactly one continuous cycle. Suppose that there is a non-zero **even** number of predefined pairs with one point in A and the other in B. Then there exists an optimal solution where no path has an edge connecting A to B. Furthermore, if $s > \frac{3}{2}$ then all optimal solutions must satisfy this property.*

Proof. To prove this we first show that in the optimal solution the endpoint cycle is ordered such that no path has its endpoints in different sets. Then our previous theorems will imply that given the required ordering of the endpoints in the cycle, there exists an optimal solution where no path has an edge connecting A to B.

To begin, consider some optimal solution to the problem. By Theorem 3.2.6, there must exist a solution such that if a given path in the solution starts and ends in the same set then it has no edges connecting A to B and if a given path in the solution starts and ends in different sets then it has exactly one edge connecting A to B. If $s > \frac{3}{2}$ then the current solution must satisfy this property by Theorem 3.2.6 and if $s = \frac{3}{2}$ then WLOG we assume the current solution satisfies this property.

Assume that there is a path P_i in the solution that has endpoints in different sets and thus has an edge connecting A to B. WLOG assume that P_i connects $a' \in A$ to $b' \in B$. Let C denote the continuous cycle formed by alternating between the endpoints of the paths and the predefined pairs in the solution. Let us establish an arbitrary traversal direction along C, against which the direction of all paths will be relative to. Since C is closed we know that there must be an even number of

alternations between A and B along the cycle. Because there is an even number of predefined pairs with one point in A and the other in B then the number of paths with one endpoint in A and the other in B must also be even to ensure that the total number of alternations between A and B along the cycle is even. Thus there must be another path P_j in the solution that has an edge connecting A to B.

Case 1: P_i and P_j are in the same direction. Let P_j connect $a'' \in A$ to $b'' \in B$. We can then note that in both P_i and P_j there must be an edge directly connecting from A to B and we will label these edges as $\{a^*, b^*\}$ and $\{a^\dagger, b^\dagger\}$ respectively. Then we can note the cycle goes:

$$\cdots \sim \boxed{a' \rightsquigarrow a^* \rightarrow b^* \rightsquigarrow b'} \sim \cdots \sim \boxed{a'' \rightsquigarrow a^\dagger \rightarrow b^\dagger \rightsquigarrow b''} \sim \ldots.$$

It follows that we can break apart the solution at the two edges $\{a^*, b^*\}$ and $\{a^\dagger, b^\dagger\}$ and reverse the order of the cycle and paths along the cutout section between these two edges to get a new solution where the cycle goes:

$$\cdots \sim \boxed{a' \rightsquigarrow a^* \rightarrow a^\dagger \rightsquigarrow a''} \sim \cdots \sim \boxed{b' \rightsquigarrow b^* \rightarrow b^\dagger \rightsquigarrow b''} \sim \ldots.$$

Because the predefined pairs are simply undirected pairings of endpoints, traversing them in reverse during this cycle modification is entirely valid and maintains all predefined pairings and does not create multiple cycles.

Since the previous solution was assumed optimal we know that $|a^*b^*| + |a^\dagger b^\dagger| \leq |a^*a^\dagger| + |b^*b^\dagger|$. Since A and B are s-well-separated we know that $\forall i, j, k; |a_i a_j| \leq \frac{1}{s}|a_i b_k|$ (and vice versa

for B). From this we can conclude:

$$\begin{aligned}|a^*b^*| + |a^\dagger b^\dagger| &\leq |a^*a^\dagger| + |b^*b^\dagger| \\ &\leq \frac{1}{s}|a^*b^*| + \frac{1}{s}|a^\dagger b^\dagger| \\ &\leq \frac{1}{s}(|a^*b^*| + |a^\dagger b^\dagger|)\end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which is a contradiction since $s \geq \frac{3}{2}$. Therefore there cannot be two paths in the same direction that have edges connecting A to B.

Case 2: P_i and P_j are in different directions. Let P_j connect $b'' \in B$ to $a'' \in A$. By Case 1 it follows that these must be the only two paths crossing between A and B in the solution since if there were any more we could find two in the same direction.

We can then note that in P_i there must be an edge directly connecting from A to B and we will label this edge as $\{a^*, b^*\}$ and in P_j there must be an edge directly connecting from B to A and we will label this edge as $\{b^\dagger, a^\dagger\}$. By linking a^* to $a^\dagger$ and b^* to $b^\dagger$ we can break apart the solution at the two edges $\{a^*, b^*\}$ and $\{b^\dagger, a^\dagger\}$ and create two disjoint cycles based on the structure of the original cycle. Let C_A be the sub-cycle that contains a^* and $a^\dagger$ (and consequently a' and a'') and let C_B be the sub-cycle that contains b^* and $b^\dagger$ (and consequently b' and b''). Since all alternations must be done via predefined pairs, at least one of these two sub-cycles must contain a predefined pair of endpoints with one endpoint in A and the other in B. WLOG assume that C_A contains such a predefined pair of endpoints (otherwise C_B would contain it and the definition of A and B could be flipped).

Let (a_1, b_1) be the first predefined pair of endpoints in C_A with $a_1 \in A$ and $b_1 \in B$ after a^* in the cycle order of C_A. Let $b_2 \in B$ be the endpoint on the other end of the path that begins with

b_1 (since b' and b'' are strictly contained within C_B, the endpoint $b_1 \in C_A$ must belong to a path other than P_i and P_j which represent the only two paths with endpoints in different sets). We can choose any edge from the b_1 to b_2 path and it will always connect two points in B by the previous application of Theorem 3.2.6. Considering the first edge in the b_1 to b_2 path, it follows that the first edge must connect b_1 to some point in B and we will label this point b_3.

We can then break apart the solution at the edge $\{b_1, b_3\}$ and create a new solution by adding $\{b_1, b^*\}$, $\{b^\dagger, b_3\}$, and $\{a^*, a^\dagger\}$ all back in. This new solution maintains all predefined pairs and forms a single cycle. To rigorously verify this, note that removing $\{a^*, b^*\}$, $\{b^\dagger, a^\dagger\}$, and $\{b_1, b_3\}$ breaks the original cycle into three disjoint directed segments: S_1 from $b_3 \rightsquigarrow a^*$, S_2 from $b^* \rightsquigarrow b^\dagger$, and S_3 from $a^\dagger \rightsquigarrow b_1$. Replacing the cut edges with $\{b_1, b^*\}$, $\{b^\dagger, b_3\}$, and $\{a^*, a^\dagger\}$ recombines these segments into the sequence:

$$S_1 \xrightarrow{\{a^*,a^\dagger\}} S_3 \xrightarrow{\{b_1,b^*\}} S_2 \xrightarrow{\{b^\dagger,b_3\}} S_1.$$

Because the new topology visits all three segments exactly once before returning to the start, it forms a single continuous cycle. Furthermore, since none of S_1, S_2, and S_3 are created by cutting through any predefined pair of endpoints, the new solution maintains all predefined pairs of endpoints.

Since the previous solution was assumed optimal we know that $|a^*b^*| + |b^\dagger a^\dagger| + |b_1 b_3| \leq |a^* a^\dagger| + |b_1 b^*| + |b^\dagger b_3|$. Define $\omega = \delta(A, B)$. The following are true by the definition of well-separated pairs and non-negativity of distances:

$$|a^*b^*| \geq \omega \quad |b^\dagger a^\dagger| \geq \omega \quad |b_1 b_3| \geq 0 \quad |a^* a^\dagger| \leq \frac{1}{s}\omega \quad |b_1 b^*| \leq \frac{1}{s}\omega \quad |b^\dagger b_3| \leq \frac{1}{s}\omega.$$

From these we can conclude: $2\omega \leq |a^*b^*| + |b^\dagger a^\dagger| + |b_1 b_3| \leq |a^* a^\dagger| + |b_1 b^*| + |b^\dagger b_3| \leq \frac{3}{s}\omega$ which implies that $s \leq \frac{3}{2}$. If $s > \frac{3}{2}$ then this is a contradiction and thus the original solution must not have had two paths with endpoints in different sets. It immediately follows from this and the previous application of Theorem 3.2.6 that there are no edges connecting A to B in the current solution as desired. If $s = \frac{3}{2}$ then we can note that all these inequalities must be tight and thus since the only two edges crossing between A and B were removed from the original solution to create the new solution, it follows that the new solution must also be optimal and has no paths with edges connecting A to B. □

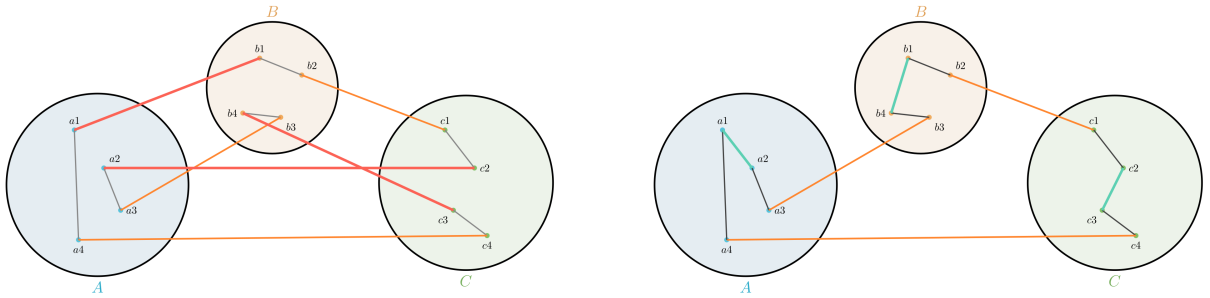
Chapter 4: Multiple Well-Separated Sets are Path/Tour Separated

In this chapter, we will show that if we have multiple well-separated sets of points, then solutions to shortest path/tour problems over these points must partition respect this structure.

4.1 Three Well-Separated Sets are Path/Tour Separated

The first extension of the results from the previous chapter is to show that if we have three pairwise well-separated sets, then the optimal paths must still respect this structure. While the proof techniques remain consistent with the simpler two-set case, the analysis becomes significantly more complex due to the increased number of cases involved. One of these cases whose analysis is unique is when the sets are connected between each other by an extra cycle across all three. In this case, we apply a similar technique to the one used in Theorems 3.1.1 and 3.1.2 to show that an extra cycle cannot be part of the optimal path/tour (this is further illustrated in fig. 4.1).

Theorem 4.1.1. *Let $P = A \cup B \cup C$ be a set of points where A, B, C are pairwise s-well-separated with $s \geq 1$. Then there exists an optimal Hamiltonian path which starts on a point $a' \in A$ and ends on a point $a'' \in A$ and must have only three or four edges connecting the three sets. Furthermore if $s > 1$ then all optimal solutions must satisfy this property.*



(a) A tour which completes two cycles across the three sets.

(b) A tour with only one cycle across the three sets.

Figure 4.1: The tour with one less cycle across the three sets must be shorter by the WSP condition.

Proof. To prove this we can observe two different cases. First consider some optimal Hamiltonian path on P with endpoints a' and a''.

Case 1: The path contains at least one edge connecting each pair of sets. WLOG let these edges be $(a_1, b_1), (b_2, c_1), (c_2, a_2)$ ordered as they appear in the path (a looped ordering is guaranteed up to the swapping of B and C). Assume that the optimal Hamiltonian path from a' to a'' over P has more than four edges connecting the three sets. Then there must either be another loop connecting the three sets or there must be at least two more edges connecting one of the pairs of sets.

Subcase 1a: There is another loop connecting the three sets both in the same orientation. Then the optimal path must contain at least six edges connecting the three sets. Let these edges be $(a_1, b_1), (b_2, c_1), (c_2, a_2), (a_3, b_3)^*, (b_4, c_3)^*, (c_4, a_4)^*$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to c_1 \rightsquigarrow c_2 \to a_2 \rightsquigarrow a_3 \to b_3 \rightsquigarrow b_4 \to c_3 \rightsquigarrow c_4 \to a_4 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 3-optimality condition implying the following: $|a_1b_1| + |c_2a_2| + |b_4c_3| \leq |a_1a_2| + |b_4b_1| + |c_2c_3|$. This is to say the following path must be at least as long:

$$a' \rightsquigarrow a_1 \rightarrow a_2 \rightsquigarrow a_3 \rightarrow b_3 \rightsquigarrow b_4 \rightarrow b_1 \rightsquigarrow b_2 \rightarrow c_1 \rightsquigarrow c_2 \rightarrow c_3 \rightsquigarrow c_4 \rightarrow a_4 \rightsquigarrow a''$$

We can then note the following from the inequality and Theorem 2.3.4:

$$\begin{aligned} |a_1b_1| + |c_2a_2| + |b_4c_3| &\leq |a_1a_2| + |b_4b_1| + |c_2c_3| \\ &\leq \frac{1}{s}|a_1b_1| + \frac{1}{s}|c_2a_2| + \frac{1}{s}|b_4c_3| \\ &= \frac{1}{s}(|a_1b_1| + |c_2a_2| + |b_4c_3|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s = 1$ then this implies that $|a_1b_1| + |c_2a_2| + |b_4c_3| = |a_1a_2| + |b_4b_1| + |c_2c_3|$ and therefore the new path, with one less loop, is also optimal.

Subcase 1b: There is another loop connecting the three sets in the opposite orientation. Then the optimal path must contain at least six edges connecting the three sets. Let these edges be $(a_1, b_1), (b_2, c_1), (c_2, a_2), (a_3, c_3)^*, (c_4, b_3)^*, (b_4, a_4)^*$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_1 \rightarrow b_1 \rightsquigarrow b_2 \rightarrow c_1 \rightsquigarrow c_2 \rightarrow a_2 \rightsquigarrow a_3 \rightarrow c_3 \rightsquigarrow c_4 \rightarrow b_3 \rightsquigarrow b_4 \rightarrow a_4 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 3-optimality condition implying the

following: $|a_1b_1| + |c_2a_2| + |c_4b_3| \leq |a_1a_2| + |c_4c_2| + |b_1b_3|$. This is to say the following path must be at least as long:

$$a' \rightsquigarrow a_1 \rightarrow a_2 \rightsquigarrow a_3 \rightarrow c_3 \rightsquigarrow c_4 \rightarrow c_2 \rightsquigarrow c_1 \rightarrow b_2 \rightsquigarrow b_1 \rightarrow b_3 \rightsquigarrow b_4 \rightarrow a_4 \rightsquigarrow a''$$

We can then note the following from the inequality and Theorem 2.3.4:

$$\begin{aligned} |a_1b_1| + |c_2a_2| + |c_4b_3| &\leq |a_1a_2| + |c_2c_4| + |b_1b_3| \\ &\leq \frac{1}{s}|a_1b_1| + \frac{1}{s}|c_2a_2| + \frac{1}{s}|c_4b_3| \\ &= \frac{1}{s}(|a_1b_1| + |c_2a_2| + |c_4b_3|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s = 1$ then this implies that $|a_1b_1| + |c_2a_2| + |c_4b_3| = |a_1a_2| + |c_4c_2| + |b_1b_3|$ and therefore the new path, with one less loop, is also optimal.

Subcase 1c: There must be at least two additional edges at the end causing a loop from A to B. Then the optimal Hamiltonian path must contain at least five edges connecting the three sets. Let these edges be $(a_1, b_1), (b_2, c_1), (c_2, a_2), (a_3, b_3)^*, (b_4, a_4)^*$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_1 \rightarrow b_1 \rightsquigarrow b_2 \rightarrow c_1 \rightsquigarrow c_2 \rightarrow a_2 \rightsquigarrow a_3 \rightarrow b_3 \rightsquigarrow b_4 \rightarrow a_4 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 2-optimality condition implying the following: $|a_1b_1| + |a_3b_3| \leq |a_1a_3| + |b_1b_3|$. This is to say the following path must be at least as

long:

$$a' \rightsquigarrow a_1 \to a_3 \rightsquigarrow a_2 \to c_2 \rightsquigarrow c_1 \to b_2 \rightsquigarrow b_1 \to b_3 \rightsquigarrow b_4 \to a_4 \rightsquigarrow a''$$

We can then note the following from the inequality and Theorem 2.3.4:

$$\begin{aligned} |a_1b_1| + |a_3b_3| &\leq |a_1a_3| + |b_1b_3| \\ &\leq \frac{1}{s}|a_1b_1| + \frac{1}{s}|a_3b_3| \\ &= \frac{1}{s}(|a_1b_1| + |a_3b_3|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s = 1$ then this implies that $|a_1b_1| + |a_3b_3| = |a_1a_3| + |b_1b_3|$ and therefore the new path, with one less loop, is also optimal.

Subcase 1d: There must be at least two additional edges at the end causing a loop from A to C. Then the optimal Hamiltonian path must contain at least five edges connecting the three sets. Let these edges be $(a_1, b_1), (b_2, c_1), (c_2, a_2), (a_3, c_3)^*, (c_4, a_4)^*$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to c_1 \rightsquigarrow c_2 \to a_2 \rightsquigarrow a_3 \to c_3 \rightsquigarrow c_4 \to a_4 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 2-optimality condition implying the following: $|c_2a_2| + |c_4a_4| \leq |c_2c_4| + |a_2a_4|$. This is to say the following path must be at least as long:

$$a' \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to c_1 \rightsquigarrow c_2 \to c_4 \rightsquigarrow c_3 \to a_3 \rightsquigarrow a_2 \to a_4 \rightsquigarrow a''$$

We can then note the following from the inequality and Theorem 2.3.4:

$$\begin{aligned} |c_2a_2| + |c_4a_4| &\leq |c_2c_4| + |a_2a_4| \\ &\leq \frac{1}{s}|c_2a_2| + \frac{1}{s}|c_4a_4| \\ &= \frac{1}{s}(|c_2a_2| + |c_4a_4|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s = 1$ then this implies that $|c_2a_2| + |c_4a_4| = |c_2c_4| + |a_2a_4|$ and therefore the new path, with one less loop, is also optimal.

Subcase 1e: There must be at least two additional edges connecting A, B before starting the primary loop. Then the optimal Hamiltonian path must contain at least five edges connecting the three sets. Let these edges be $(a_3, b_3)^*, (b_4, a_4)^*, (a_1, b_1), (b_2, c_1), (c_2, a_2)$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_3 \to b_3 \rightsquigarrow b_4 \to a_4 \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to c_1 \rightsquigarrow c_2 \to a_2 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 2-optimality condition implying the following: $|a_3b_3| + |a_1b_1| \leq |a_3a_1| + |b_3b_1|$. This is to say the following path must be at least as long:

$$a' \rightsquigarrow a_3 \to a_1 \rightsquigarrow a_4 \to b_4 \rightsquigarrow b_3 \to b_1 \rightsquigarrow b_2 \to c_1 \rightsquigarrow c_2 \to a_2 \rightsquigarrow a''$$

We can then note the following from the inequality and Theorem 2.3.4:

$$\begin{aligned}|a_3b_3| + |a_1b_1| &\leq |a_3a_1| + |b_3b_1| \\ &\leq \frac{1}{s}|a_3b_3| + \frac{1}{s}|a_1b_1| \\ &= \frac{1}{s}(|a_3b_3| + |a_1b_1|)\end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s = 1$ then this implies that $|a_3b_3| + |a_1b_1| = |a_3a_1| + |b_3b_1|$ and therefore the new path, with one less loop, is also optimal.

Subcase 1f: There must be at least two additional edges connecting A, C before starting the primary loop. Then the optimal Hamiltonian path must contain at least five edges connecting the three sets. Let these edges be $(a_3, c_3)^*, (c_4, a_4)^*, (a_1, b_1), (b_2, c_1), (c_2, a_2)$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_3 \to c_3 \rightsquigarrow c_4 \to a_4 \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to c_1 \rightsquigarrow c_2 \to a_2 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 2-optimality condition implying the following: $|c_4a_4| + |c_2a_2| \leq |c_4c_2| + |a_4a_2|$. This is to say the following path must be at least as long:

$$a' \rightsquigarrow a_3 \to c_3 \rightsquigarrow c_4 \to c_2 \rightsquigarrow c_1 \to b_2 \rightsquigarrow b_1 \to a_1 \rightsquigarrow a_4 \to a_2 \rightsquigarrow a''$$

We can then note the following from the inequality and Theorem 2.3.4:

$$\begin{aligned}|c_4a_4| + |c_2a_2| &\leq |c_4c_2| + |a_4a_2| \\ &\leq \frac{1}{s}|c_4a_4| + \frac{1}{s}|c_2a_2| \\ &= \frac{1}{s}(|c_4a_4| + |c_2a_2|)\end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s = 1$ then this implies that $|c_4a_4| + |c_2a_2| = |c_4c_2| + |a_4a_2|$ and therefore the new path, with one less loop, is also optimal.

Subcase 1g: There is a loop back from B to C before the primary loop is ended. Then the optimal Hamiltonian path must contain at least five edges connecting the three sets. Let these edges be $(a_1, b_1), (b_2, c_1), (c_4, b_3)^*, (b_4, c_3)^*, (c_2, a_2)$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to c_1 \rightsquigarrow c_4 \to b_3 \rightsquigarrow b_4 \to c_3 \rightsquigarrow c_2 \to a_2 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 2-optimality condition implying the following: $|b_2c_1| + |b_4c_3| = |b_2b_4| + |c_1c_3|$. This is to say the following path must be at least as long:

$$a' \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to b_4 \rightsquigarrow b_3 \to c_4 \rightsquigarrow c_1 \to c_3 \rightsquigarrow c_2 \to a_2 \rightsquigarrow a''$$

We can then note the following from the inequality and Theorem 2.3.4:

$$\begin{aligned}|b_2c_1| + |b_4c_3| &\leq |b_2b_4| + |c_1c_3| \\ &\leq \frac{1}{s}|b_2c_1| + \frac{1}{s}|b_4c_3| \\ &= \frac{1}{s}(|b_2c_1| + |b_4c_3|)\end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s = 1$ then this implies that $|b_2c_1| + |b_4c_3| = |b_2b_4| + |c_1c_3|$ and therefore the new path, with one less loop, is also optimal.

Case 2: The path does not pairwise connect each of the three sets.

Subcase 2a: A is only connected to a single other set. WLOG let the only edges connecting sets be from A to B and B to C (C and B can be alternated symmetrically). Then the path must contain at least four edges connecting the three sets. Let these edges be $(a_1, b_1), (b_2, c_1), (c_2, b_3), (b_4, a_2)$ ordered as they appear in the path.

Subsubcase 2a.i: There is an extra loop connecting A to B before/after the main loop. WLOG assume that the extra loop occurs before the main loop (if not then the path can be reversed and preserve the same structure). Then the optimal Hamiltonian path must contain at least six edges connecting the three sets. Let these edges be $(a_3, b_5)^*, (b_6, a_4)^*, (a_1, b_1), (b_2, c_1), (c_2, b_3), (b_4, a_2)$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_3 \to b_5 \rightsquigarrow b_6 \to a_4 \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to c_1 \rightsquigarrow c_2 \to b_3 \rightsquigarrow b_4 \to a_2 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 2-optimality condition implying the following: $|a_3b_5| + |a_1b_1| \leq |a_3a_1| + |b_5b_1|$. This is to say the following path must be at least as

long:

$$a' \rightsquigarrow a_3 \to a_1 \rightsquigarrow a_4 \to b_6 \rightsquigarrow b_5 \to b_1 \rightsquigarrow b_2 \to c_1 \rightsquigarrow c_2 \to b_3 \rightsquigarrow b_4 \to a_2 \rightsquigarrow a''$$

$$\begin{aligned} |a_3b_5| + |a_1b_1| &\leq |a_3a_1| + |b_5b_1| \\ &\leq \frac{1}{s}|a_3b_5| + \frac{1}{s}|a_1b_1| \\ &= \frac{1}{s}(|a_3b_5| + ||a_1b_1|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s = 1$ then this implies that $|a_3b_5| + |a_1b_1| = |a_3a_1| + |b_5b_1|$ and therefore the new path, with one less loop, is also optimal.

Subsubcase 2a.ii: There is an extra loop connecting B to C before ending the path. Then the optimal Hamiltonian path must contain at least six edges connecting the three sets. WLOG let these edges be $(a_1, b_1), (b_2, c_1), (c_3, b_5)^*, (b_6, c_4)^*, (c_2, b_3), (b_4, a_2)$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to c_1 \rightsquigarrow c_3 \to b_5 \rightsquigarrow b_6 \to c_4 \rightsquigarrow c_2 \to b_3 \rightsquigarrow b_4 \to a_2 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 2-optimality condition implying the following: $|b_2c_1| + |b_6c_4| \leq |b_2b_6| + |c_1c_4|$. This is to say the following path must be at least as long:

$$a' \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to b_6 \rightsquigarrow b_5 \to c_3 \rightsquigarrow c_1 \to c_4 \rightsquigarrow c_2 \to b_3 \rightsquigarrow b_4 \to a_2 \rightsquigarrow a''$$

We can then note the following from the inequality and Theorem 2.3.4:

$$\begin{aligned}|b_2c_1| + |b_6c_4| &\leq |b_2b_6| + |c_1c_4| \\ &\leq \frac{1}{s}|b_2c_1| + \frac{1}{s}|b_6c_4| \\ &= \frac{1}{s}(|b_2c_1| + |b_6c_4|)\end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s = 1$ then this implies that $|a_3b_5| + |a_1b_1| = |a_3a_1| + |b_5b_1|$ and therefore the new path, with one less loop, is also optimal.

Subcase 2b: There are no connections between B and C WLOG let these edges be $(a_1, b_1), (b_2, a_2), (a_3, c_1)$, ordered as they appear in the path (if C occurs before B then flip the names).

Subsubcase 2b.i: There is an extra loop connecting A to B before / after the main loop. WLOG assume that the extra loop occurs before the main loop (the opposite case is symmetric). Then the optimal Hamiltonian path must contain at least six edges connecting the three sets. Let these edges be $(a_3, b_5)^*, (b_6, a_4)^*, (a_1, b_1), (b_2, a_2), (a_5, c_1), (c_2, a_6)$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_5 \to a_3 \rightsquigarrow a_4 \to b_6 \rightsquigarrow b_2 \to a_2 \rightsquigarrow a_5 \to c_1 \rightsquigarrow c_2 \to a_6 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 2-optimality condition implying the following: $|a_1b_1| + |a_4b_6| \leq |a_1a_4| + |b_1b_6|$. This is to say the following path must be at least as long:

$$a' \rightsquigarrow a_1 \to a_4 \rightsquigarrow a_4 \to b_5 \rightsquigarrow b_1 \to b_6 \rightsquigarrow b_2 \to a_2 \rightsquigarrow a_5 \to c_1 \rightsquigarrow c_2 \to a_6 \rightsquigarrow a''$$

We can then note the following from the inequality and Theorem 2.3.4:

$$\begin{aligned}|a_1b_1| + |a_4b_6| &\leq |a_1a_4| + |b_1b_6| \\ &\leq \frac{1}{s}|a_1b_1| + \frac{1}{s}|a_4b_6| \\ &= \frac{1}{s}(|a_1b_1| + |a_4b_6|)\end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s = 1$ then this implies that $|a_1b_1| + |a_4b_6| = |a_1a_4| + |b_1b_6|$ and therefore the new path, with one less loop, is also optimal.

Subsubcase 2b.ii: There is an extra loop connecting A to C before / after the main loop. WLOG assume that the extra loop occurs after the main loop (the opposite case is symmetric). Then the optimal Hamiltonian path must contain at least six edges connecting the three sets. Let these edges be $(a_1, b_1), (b_2, a_2), (a_3, c_1), (c_2, a_4), (a_5, c_3)^*, (c_4, a_6)^*$ ordered as they appear in the path. This gives the following partial structure of the path:

$$a' \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to a_2 \rightsquigarrow a_3 \to c_1 \rightsquigarrow c_2 \to a_4 \rightsquigarrow a_5 \to c_3 \rightsquigarrow c_4 \to a_6 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 2-optimality condition implying the following: $|a_3c_1| + |a_5c_3| \leq |a_3a_5| + |c_1c_3|$. This is to say the following path must be at least as long:

$$a' \rightsquigarrow a_1 \to b_1 \rightsquigarrow b_2 \to a_2 \rightsquigarrow a_3 \to a_5 \rightsquigarrow a_4 \to c_2 \rightsquigarrow c_1 \to c_3 \rightsquigarrow c_4 \to a_6 \rightsquigarrow a''$$

$$\begin{aligned}|a_3c_1| + |a_5c_3| &\leq |a_3a_5| + |c_1c_3| \\ &\leq \frac{1}{s}|a_3c_1| + \frac{1}{s}|a_5c_3| \\ &= \frac{1}{s}(|a_3c_1| + |a_5c_3|)\end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. If $s = 1$ then this implies that $|a_3c_1| + |a_5c_3| = |a_3a_5| + |c_1c_3|$ and therefore the new path, with one less loop, is also optimal.

Across all cases we have that $s \leq 1$. If $s > 1$ then we have a contradiction in all cases and thus all optimal paths must have at most four edges connecting the three sets as desired. If $s = 1$ then we have seen that a path with one less loop can always be created across all cases. We can then repeatedly apply the cases until a path with at most four edges connecting the three sets is reached.

Therefore there must exist an optimal path over P between a' and a'' where there are either three or four edges connecting the three sets. Furthermore, if $s > 1$ then all optimal paths must satisfy this property. □

Corollary 4.1.1.1. *Let $P = A \cup B \cup C$ be a set of points where A, B, C are pairwise s-well-separated with $s > 1$. Then A, B, and C must be TSP-separated.*

4.2 k Well-Separated Sets are Path/Tour Separated

Theorem 4.2.1. *Let $P = \bigcup_{i=1}^{n} A_i$ be a set of points where $A_1, A_2, \ldots, A_n$ are all pairwise s-well-separated with $s \geq 1$ (and $n \geq 2$). Then there exists a optimal Hamiltonian path over P which starts at $a' \in A_i$ and ends at $a'' \in A_j$ where no set is ever directly connected to another set more than twice in the same direction. If $s > 1$ then all optimal Hamiltonian paths must satisfy this*

property.

Proof. Consider some optimal Hamiltonian path over P which starts at $a' \in A_i$ and ends at $a'' \in A_j$. Assume there is two pairs of sets $B = A_k$ and $C = A_l$ which are directly connected by two edges in the same direction (where $k \neq l$). WLOG let these edges be (b_1, c_1) and (b_2, c_2) where $b_1, b_2 \in B$ and $c_1, c_2 \in C$ ordered as they appear in the path (if C occurs before B then flip the names).

This gives the following partial structure of the path:

$$a' \rightsquigarrow b_1 \to c_1 \rightsquigarrow b_2 \to c_2 \rightsquigarrow a''$$

Since the path is optimal we know that it satisfies the 2-optimality condition implying that $|b_1c_1| + |b_2c_2| \leq |b_1b_2| + |c_1c_2|$. This is to say that the following path must be at least as long as the original path:

$$a' \rightsquigarrow b_1 \to b_2 \rightsquigarrow c_1 \to c_2 \rightsquigarrow a''$$

Since C and B are s-well-separated we know that $\forall i, j, k; |b_ib_j| \leq \frac{1}{s}|b_ic_k|$ (and vice versa for C). From this we can conclude the following:

$$\begin{aligned} |b_1c_1| + |b_2c_2| &\leq |b_1b_2| + |c_1c_2| \\ &\leq \frac{1}{s}|b_1c_1| + \frac{1}{s}|b_2c_2| \\ &= \frac{1}{s}(|b_1c_1| + |b_2c_2|) \end{aligned}$$

This implies that $1 \leq \frac{1}{s}$ which in turn implies that $s \leq 1$. Thus if $s > 1$ then we have a contradiction

and therefore no optimal Hamiltonian path can have more than two edges directly connecting any pair of sets in the same direction. If $s = 1$ then we can conclude that $|b_1c_1|+|b_2c_2| = |b_1b_2|+|c_1c_2|$ and therefore the second path is also an optimal Hamiltonian path over P. We can repeat this process over all pairs until every set is connected by no more than one edge to another set in a given direction.

Therefore there exists an optimal Hamiltonian path over P which starts at $a' \in A_i$ and ends at $a'' \in A_j$ where no set is ever directly connected to another set more than twice in the same direction. □

Chapter 5: Partitioning Algorithms

The previous sections have focused on what can be done once vertices have been partitioned into well-separated sets. Although extensive research exists on calculating WSPDs [HRR25; GZ05; Cha08; CK95; PM13; Gov+06], some approaches are limited to specific metric spaces and many prioritize minimizing the total number of pairs—an objective distinct from our own. Specifically, these algorithms fall short in two regards that we focus on: firstly, we seek to maximize the balance of the partitioning, and secondly, we may seek to limit the number of pairs at a given level of the WSPD.

We therefore propose two algorithms in this section that are designed to focus on these two new goals. The first is a greedy algorithm which, for a given set of vertices, identifies the most balanced partitioning into two well-separated sets. The second algorithm handles the more complex case of partitioning into up to k well-separated sets and is designed to minimize the largest number of points in any given set. These algorithms operate by iterating through all valid separation distances to determine if cutting a specific edge yields a valid partition and, subsequently, calculating its balance. While these methods have a significantly higher computational cost than standard WSPD algorithms, they ensure that the resulting partitions maximize the potential for complexity reduction.

5.1 Bipartitions

While it is well-established that a WSPD can be used to compute the geometric minimum spanning tree (MST) [NZ02], the converse also holds: the MST of a set of points can be used to efficiently derive balanced WSPs and thus a decomposition. A similar result was first shown by Gower and Ross [GR69], who demonstrated that the MST of a set of points can be used to calculate a single-linkage cluster analysis (SLCA)—a hierarchical clustering of the points based on the distance between them. In this section, we demonstrate that for two s-well-separated sets of points, the minimum distance between the two sets exists within the MST of the points. We then describe a greedy algorithm for calculating a bipartition of a set of points into two well-separated sets based on this result. We then prove the correctness of this algorithm for $s > 1$ and analyze its runtime. Finally, we show that for $s \leq 1$ the problem of finding the best balanced bipartition is NP-hard. It should be noted that at non-zero separation factors, it may be impossible to find a bipartition of a set of points into two well-separated sets.

Lemma 5.1.1. *Let (A, B) be a bi-partition of a set of points P within a metric space such that $P = A \cup B$. Then any MST of P must contain at least one edge connecting the two clusters A and B that is minimal among all edges connecting A and B.*

Proof. Let $e = \{u, v\}$ be an edge representing the minimum distance between A and B where $u \in A$ and $v \in B$. Suppose that e is not in some MST of P. Then there must exist another edge $e' = \{u', v'\}$ in the cycle where $u' \in A$ and $v' \in B$ otherwise we wouldn't have a spanning tree. Then by adding e to the MST we create a cycle. Since e represents the minimum distance between A and B we have that $\omega(e) \leq \omega(e')$. However by the minimax property of the MST we know that

$\omega(e') \leq \omega(e)$. Thus $\omega(e) = \omega(e')$ and therefore the MST contains an edge connecting A and B that is minimal among all edges connecting A and B. $\square$

Algorithm 1 BESTBALANCEDBIPARTITION(D, s)

Require: A metric distance matrix $D \in \mathbb{R}^{n \times n}$, separation factor $s > 1$

Ensure: A bi-partition (A, B) that is s-well-separated and maximally balanced, or NULL if no such partition exists

$E \leftarrow$ BUILDMST(D) ▷ $n - 1$ *edges*
bestScore $\leftarrow +\infty$
bestPartition $\leftarrow$ NULL
for all edges $e = (u, v)$ in E **do**
    $(A, B) \leftarrow$ CONNECTEDCOMPONENTS$(E \setminus \{e\})$
    balance $\leftarrow \max(|A|, |B|)$ ▷ *Balance score: size of larger cluster*
    if balance $\geq$ bestScore **then**
        continue ▷ *Skip if not better than current best*
    $\delta \leftarrow w(e)$ ▷ *Single-linkage separation distance*
    $\mathrm{diam}(A) \leftarrow$ COMPUTEDIAMETER(A, D)
    $\mathrm{diam}(B) \leftarrow$ COMPUTEDIAMETER(B, D)
    $M \leftarrow \max\{\mathrm{diam}(A), \mathrm{diam}(B)\}$
    if $\delta \geq s \cdot M$ **then** ▷ s*-well-separated condition*
        bestScore $\leftarrow$ balance
        bestPartition $\leftarrow (A, B)$
        if bestScore $= \lceil n/2 \rceil$ **then** ▷ *Perfectly balanced partition found*
            break
return bestPartition

Analyzing the program's complexity: calculating the MST takes $O(n^2)$ time and yields $n-1$ edges. For each edge, we perform three steps: (1) compute the connected components in $O(n)$ time, (2) compute the diameter of each component in $O(n^2)$ time, and (3) check the separation condition in $O(1)$ time. Consequently, the total runtime is $O(n^3)$.

Lemma 5.1.2. *Let (A, B) be a bi-partition of a set of points P within a metric space. Suppose that more than one cut of any MST is required to separate A and B. Then (A, B) cannot be s-well-separated for any $s > 1$.*

Proof. Let $e = (u, v)$ be the edge in the MST representing the minimum distance between A and B where $u \in A$ and $v \in B$ which must exist by Lemma 5.1.1. Since more than one cut is required to separate A and B, there must be another edge $e' = (v', u')$ in the MST where $u' \in A$ and $v' \in B$. Let $M = \max\{\text{diam}(A), \text{diam}(B)\}$. Define $e^* = (u, u')$ then we know that $\omega(e^*) \leq M$ by the definition of diameter. However by the minimax property of the MST we have that $\omega(e) \leq \omega(e^*)$. Thus $\omega(e) \leq M$ and therefore (A, B) cannot be s-well-separated for any $s > 1$. □

Theorem 5.1.3. *If there exists a bi-partition of the points that is s-well-separated (where $s > 1$), then Algorithm 1 will find a bi-partition that is s-well-separated and maximally balanced.*

Proof. It follows from Lemma 5.1.2 that since $s > 1$ then any bi-partition that is s-well-separated must be separable by a single cut of any MST. Since Algorithm 1 iterates through all edges of the MST and checks the bi-partition induced by cutting that edge, it will find all bi-partitions that are s-well-separated. Among these bi-partitions, the algorithm keeps track of the one with the best balance score (size of larger cluster) and returns it at the end. Therefore if there exists a bi-partition that is s-well-separated, the algorithm will find and return a bi-partition that is s-well-separated and maximally balanced. □

Theorem 5.1.4. *Finding the best balanced bi-partition for $s \leq 1$ is NP-hard.*

Proof. This can be shown by a reduction from the Partition problem. Given a multiset of n positive integers $S = \{x_1, x_2, \ldots, x_n\}$, the Partition problem asks whether S can be partitioned into two subsets S_1 and S_2 such that the sum of the elements in each subset is equal. We can construct a special set of points P in the following way: for each integer x_i in S, create x_i points that are clumped on a unique vertex of the $n-1$ simplex. Assume that the solution to the partition problem has each partition made up of at least two integers (if not we can trivially check for this case). Since every clump lies on a unique vertex of the simplex, the diameter of any partition is 1 and the distance between any two partitions is 1 thus making the separation factor exactly 1. Therefore any bi-partition of P is 1-well-separated. It can then be noted that after solving for the best balanced bi-partition of P that is s-well-separated for $s \leq 1$, if the resulting bi-partition has both clusters with equal size, then the answer to the Partition problem is yes and if not then the answer must be no. □

5.2 Partitioning into k sets

In the case of $k \geq 3$ clusters it cannot be guaranteed that the minimum distance between any two clusters will be represented by an edge in the MST of the points. Therefore we must instead iterate through all possible diameter thresholds and check the resulting connected components for feasibility and balance. The structure of the algorithm from there on is largely the same. The algorithm is described in Algorithm 2. The algorithm has runtime $O(n^4 + n^2k)$ since there are $O(n^2)$ diameter thresholds and for each threshold we must: (1) call connected components in $O(n)$ time, (2) check the diameter of each component in $O(n^2 + k)$ time, (3) ensure each component is

well-separated from the others in $O(n^2 + k^2)$, and (4) calculate the balance of the partition in $O(k)$ time.

Algorithm 2 BALANCEDMETRICSPLIT(D, s, k)

Require: A metric distance matrix $D \in \mathbb{R}^{n \times n}$ (symmetric, triangle inequality), separation factor $s > 0$, and number of clusters $k \geq 2$

Ensure: A no more than k-partition $\mathcal{P} = \{S_1, \ldots, S_k\}$ that maximizes balance such that every pair of clusters is s-well-separated (or a single cluster of all points is returned)

$\mathcal{M} \leftarrow \{D_{ij} \mid 1 \leq i < j \leq n\}$ ▷ *candidate diameter thresholds*
$\mathcal{S}^\star \leftarrow \big\{\{1, 2, \ldots, n\}\big\}$; bestScore $\leftarrow n$
for all $M \in \mathcal{M}$ **do**
  $E_M \leftarrow \{(i, j) \mid 1 \leq i < j \leq n,\ D_{ij} < s \cdot M\}$
  $G_M \leftarrow (V = \{1, \ldots, n\}, E_M)$
  $\mathcal{C} \leftarrow$ CONNECTEDCOMPONENTS(G_M)
  if $1 < |\mathcal{C}| \leq k$ **then**
    continue
  ▷ *Diameter feasibility: ensure each component has diameter $\leq M$*
  ok $\leftarrow$ **true**
  for all $C \in \mathcal{C}$ **do**
    if $\mathrm{diam}(C) > M$ **then**
      ok $\leftarrow$ **false**
      break
  if not ok **then**
    continue
  ▷ *Separation feasibility: ensure pairwise separation*
  sep $\leftarrow$ **true**
  for all distinct $C, C' \in \mathcal{C}$ **do**
    $\Delta(C, C') \leftarrow \min\{D_{xy} \mid x \in C,\ y \in C'\}$
    if $\Delta(C, C') < s \cdot \max(\mathrm{diam}(C), \mathrm{diam}(C'))$ **then**
      sep $\leftarrow$ **false**
      break
  if not sep **then**
    continue
  ▷ *Balance objective: minimize size of largest cluster*
  score $\leftarrow \max_{C \in \mathcal{C}} |C|$
  if score $<$ bestScore **then**
    bestScore $\leftarrow$ score
    $\mathcal{S}^\star \leftarrow \mathcal{C}$
return $\mathcal{S}^\star$

Chapter 6: Applications of Global Separation

We have shown that for $s \geq 1$, the search space for the TSP can be significantly pruned, as only two edges can cross the bipartition. Although we can efficiently identify the most balanced bipartition for $s < 1$, this approach depends on the existence of such a bipartition for the given s. In this section, we examine various problem instances to show that such bipartitions may be found in the wild. Furthermore, we generate Euclidean balls of points that are s-well-separated for $s < 1$; our results indicate that even in these cases, optimal TSP solutions often have only two edges crossing the bipartition. We confirm these findings by analyzing the TSPLIB dataset, where we find that many problems have bipartitions that are s-well-separated for $s < 1$ and that for these problems it is often the case that only two edges cross the bipartition in some optimal TSP solution.

6.1 TSPLIB

TSPLIB is a repository created by Reinelt [Rei91] of sample instances for the TSP and acts as a standard benchmark for TSP algorithms. Due to errors in floating point precision, the rounding used in a number of distance functions from the dataset such as EUC_2D and CEIL_2D can break the properties of the metric space. Even with this issue, we can note that our results would immediately apply if the distances were computed using the EXACT_2D distance function, and if not then our results would likely still guide us to very nearly optimal solutions in the worst case.

The TSPLIB repository contains 63 metric TSP instances with fewer than 2000 points. Of those 63 instances, pr107 stands out as the only instance that has a bi-partition that is s-well-separated for some $s > 1$. The problem can then be split into two clusters of 54 and 53 points. If solved using a dynamic programming algorithm, this constitutes a major speedup. Specifically, taking the ratio of the approximate number of operations we get

$$\frac{2^{107}107^2}{2\binom{54}{2}\binom{53}{2}(2^{54}54^2 + 2^{53}53^2)} \approx 10^9.$$

6.2 Varying Separation Factor

Although we have demonstrated numerous instances where our global heuristic is applicable, many problems lack a bipartition for $s \geq 1$. This threshold of $s \geq 1$ is derived from our theorems, which guarantee that if such a bipartition exists, we can find it efficiently and only two edges cross the bipartition in some optimal TSP solution. Nevertheless, many problem instances have bipartitions that are s-well-separated for some $s < 1$, and in these cases, we ask the question: is it still the case that only two edges cross the bipartition in some optimal TSP solution?

6.2.1 Point Generation

To answer this question, we construct test instances by sampling $2n$ points distributed equally across two opposing semicircles of radius 1. The semicircles are placed at a distance s from one another to ensure they are s-well-separated. We then compute optimal TSP solutions using Concorde [App+03] to determine whether more than two edges cross the bipartition. Our experiment comprises 10,000 trials for each configuration, covering separation factors $s \in [0.0, 0.5]$ in increments

of 0.05 and point counts $n \in [5, 40]$ in increments of 5.

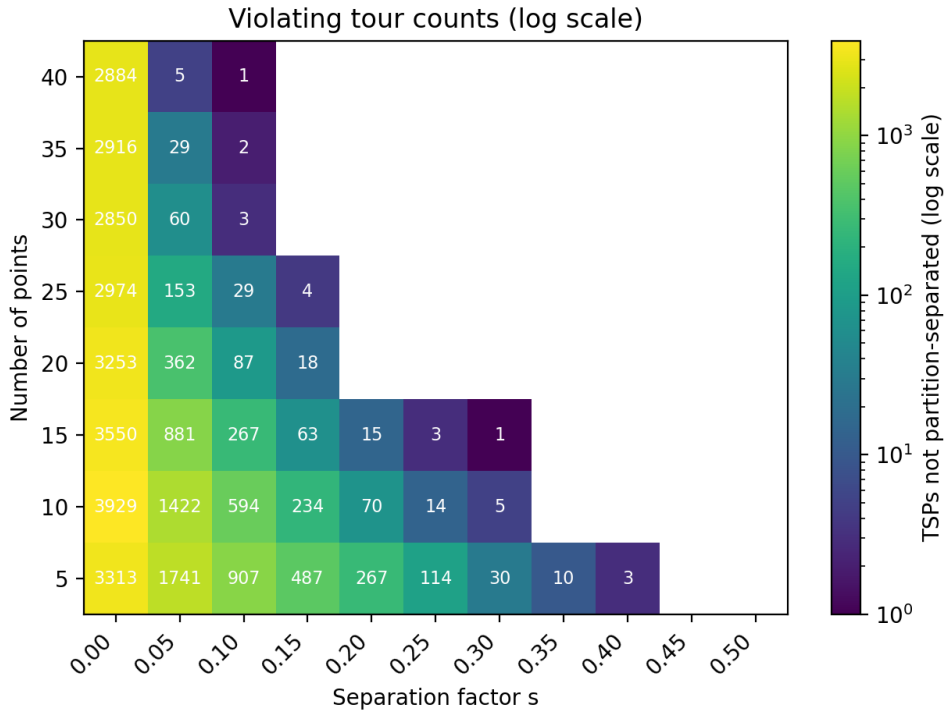


Figure 6.1: Count of problems out of 10,000 generated that had more than two edges crossing the bipartition in the optimal TSP solution.

The results of this experiment are displayed in fig. 6.1. We see that for $s \geq 0.45$, there were no problems with more than two edges crossing the bipartition in the optimal solution out of 160,000 generated problems across all values of n and $s \geq 0.45$. There is a clear trend in the data: as both s and n increase, the number of problems with more than two crossing edges in the optimal solution decreases exponentially. These results extend even to problems with a separation factor as low as $s = 0.05$, where if $n \geq 25$, fewer than 1.5% of the generated problems have more than two crossing edges. These results strongly suggest that the ”two-edge crossing” property often holds for bipartitions with $s < 1$, providing a viable heuristic even when the $s \geq 1$ theoretical threshold is not met.

6.2.2 Verification on TSPLIB

To verify that these results extend to real-world problems, we also ran the same experiment on the TSPLIB dataset. For each problem with fewer than 1000 points we first run Algorithm 1 to find a balanced bipartition that is s-well-separated for $s = 0.1$. We find that out of the 48 smaller ΔTSP instances in the dataset, there are 34 instances for which the algorithm finds a bipartition. It should be noted that, as discussed in section 5.1, it is possible that a bipartition exists; however, our algorithm fails to find it since it is not guaranteed to find a bipartition that is s-well-separated for $s < 1$. For each of the 34 instances, we then solve the TSP optimally using Concorde. Finally, we count the number of edges crossing the bipartition in the optimal solution and find that for 30 out of the 34 instances, there are only two edges crossing the bipartition. These results show that while problems with s-well-separated bipartitions for $s \geq 1$ may be rare, there are many problems with s-well-separated bipartitions for $s < 1$. Second, while there may exist counterexamples to the claim that bipartitions for $s < 1$ have only two edges crossing in the optimal solution, these counterexamples appear to be rare in practice.

Chapter 7: General Heuristics Given a WSPD

While a ΔTSP can be broken apart if there exists a bipartition that is s-well-separated for some $s \geq 1$, many problems do not have such a bipartition and thus the methods of chapter 6 cannot be guaranteed. Furthermore, even when such a bipartition exists, the two smaller problems may still be too large to solve optimally, and thus we need heuristics to solve the smaller problems and examine what happens on a pair by pair basis. A number of common heuristics and approximation techniques have been proposed for the TSP, such as the nearest neighbor heuristic [HW15], the Christofides algorithm [Chr22], k-opt [Kre89; HHH24], ant colony optimization [Mur+23], simulated annealing [BŞ13; JK91; dSFA21], and genetic algorithms [Sch18; Tsa+14]. The most popular and well-known of all heuristics for the TSP is the Lin-Kernighan-Helsgaun (LKH) heuristic [LK73; Hel09; THW18], which is a variable k-opt method that has been shown to produce near-optimal solutions for large TSP instances in practice. Most heuristics for the tsp fix problems by taking a given tour and then making local modifications to it, such as by swapping edges or by reordering the tour in some way. In the same sense we postulate that the structure of WSPDs can be used to identify areas of interest for modification to a given tour, and thus can be used to guide heuristics for improving said tour.

7.1 Identification Algorithm

To build such a heuristic, we leverage the results from section 3.5 to identify areas of interest for modification to a given tour. Given a pair $\{A, B\}$ of well-separated sets from a WSPD, we can classify into one of the three following cases based on the way in which the tour enters and exits the sets:

1. Whenever the tour exits $A \cup B$ via A (resp. B) it re-enters via A (resp. B)
2. Whenever the tour exits $A \cup B$ via A (resp. B) it re-enters via B (resp. A) an *odd* number of times
3. Whenever the tour exits $A \cup B$ via A (resp. B) it re-enters via B (resp. A) a strictly positive *even* number of times

Each of these cases can be covered by each of the following theorems which bound the number of edges between A and B in an optimal TSP tour when $s \geq \frac{3}{2}$.

1. By Theorem 3.5.1 there must be no more than two edges between A and B in an optimal TSP tour.
2. By Theorem 3.5.2 there must be no more than one edge between A and B in an optimal TSP tour.
3. By Theorem 3.5.3 there must be no edges between A and B in an optimal TSP tour.

It immediately follows that given a ΔTSP instance and a s-WSPD for $s > 1.5$ of the instance, any violation immediately identifies an area of interest which is guaranteed to improve the tour if

modified. Furthermore following the same intuition from section 6.2, we hypothesize that even for $s < 1.5$ reexamining the tour within these areas of interest will often yield improvements to the tour, and thus this method can be used as a heuristic for improving a given tour even when the theoretical threshold is not met.

To efficiently identify these areas of interest we develop Algorithm 3. Our algorithm takes in two lists of positions within the tour P_A, P_B and the number of nodes in the tour n, and returns a boolean indicating whether the tour violates the conditions for the given pair $\{A, B\}$. The algorithm may be trivially modified to directly take in A, B and a mapping of nodes to tour positions P. It may also be modified to return the specific edges which cross between A and B causing the violation. This check is very efficient and runs in $O(|P_A| + |P_B|)$ time, which is at most $O(n)$ but is often much smaller in practice and has space complexity of $O(1)$ (or $O(|P_A| + |P_B|)$ if we want to return the specific edges which violate the conditions).

Algorithm 3 WSP_HEURISTIC_CHECK(P_A, P_B, n)

Require: P_A, P_B: Sorted arrays of positions within the current tour for sets A and B.

Require: $|A| > 0, |B| > 0$ and $P_A \cap P_B = \emptyset$.

Require: n: Total number of nodes in the tour.

Ensure: Boolean indicating if the path satisfies WSP conditions.

$c_{AB} \leftarrow 0, c_{BA} \leftarrow 0$ ▷ *Direct edge counters*
$x_{cross} \leftarrow 0$ ▷ *Cross-set exit counter*
$i \leftarrow 0, j \leftarrow 0$
$k \leftarrow |P_A| + |P_B|$
for $step = 0 \rightarrow k$ **do** ▷ *Loop $k+1$ times to handle wrap-around*
  if $step = k$ **then**
    $v_{pos} \leftarrow first_{pos}$
    $v_{set} \leftarrow first_{set}$
  else if $j = |P_B|$ **or** $(i < |P_A|$ **and** $P_A[i] < P_B[j])$ **then**
    $v_{pos} \leftarrow P_A[i]$
    $v_{set} \leftarrow A$
    $i \leftarrow i + 1$
  else
    $v_{pos} \leftarrow P_B[j]$
    $v_{set} \leftarrow B$
    $j \leftarrow j + 1$
  if $step = 0$ **then**
    $first_{pos} \leftarrow v_{pos}$
    $first_{set} \leftarrow v_{set}$
  else
    $\Delta \leftarrow (v_{pos} - u_{pos} + n) \bmod n$ ▷ *Distance forward in tour*
    if $\Delta = 1$ **then** ▷ *Directly connected*
      if $u_{set} = A$ **and** $v_{set} = B$ **then** $c_{AB} \leftarrow c_{AB} + 1$
      else if $u_{set} = B$ **and** $v_{set} = A$ **then** $c_{BA} \leftarrow c_{BA} + 1$
    else if $\Delta > 1$ **and** $u_{set} \neq v_{set}$ **then** ▷ *Exit spans sets*
      $x_{cross} \leftarrow x_{cross} + 1$
  $u_{pos} \leftarrow v_{pos}$ ▷ *Advance current node*
  $u_{set} \leftarrow v_{set}$
▷ *Evaluate conditions based on cross exits*
if $x_{cross} = 0$ **then**
  return $(c_{AB} = 1$ **and** $c_{BA} = 1)$
else if $x_{cross} \bmod 2 = 0$ **then**
  return $(c_{AB} = 0$ **and** $c_{BA} = 0)$
else
  return $(c_{AB} + c_{BA} = 1)$

7.2 Methodology

To evaluate the effectiveness of this heuristic, we take the following approach. For a given value of $s \in \{0.25, 0.5, 0.75, 1.0, 1.25, 1.5\}$, we first generate a s-WSPD over a given problem instance. Then for each pair $\{A, B\}$ in the WSPD, we check whether the tour violates the conditions for that pair using Algorithm 3. We can note that building the WSPD takes $O(n \log n + ns^d)$ time (for euclidean instances), and outputs $O(s^d n)$ pairs, and then for each pair we can check for violations in $O(n)$ time, thus the overall time complexity of flagging violating pairs is $O(n \log n + s^d n^2)$.

Once we have identified a violating pair $\{A, B\}$, we disconnect all edges containing a points in $A \cup B$ and then consider how best to add back in the points of A and B to the tour. If the size of the pair is small enough say $|A| + |B| < 25$, then we can resolve the TSP by cutting all edges with a node in $A \cup B$ and reconnect the tour using a dynamic programming approach — similar to that of the traditional TSP algorithm by Held and Karp [HK61] and Bellman [Bel62] — which will find the optimal reconstruction of the tour adding back in the nodes of $A \cup B$. If the size of the pair is too large for this approach, then we reformulate the reconstruction as a TSP itself. We do this by first taking all entrance and exit points of the non-$A \cup B$ portion of the tour and then adding in the points of A and B to create a new TSP instance. We then set the distance of any entrance/exit pairs to be zero and add a large constant penalty to all other distances to ensure that the solution respects the rules of the problem (specifically that an entrance node to the non-$A \cup B$ portion of the tour must be immediately followed by an exit node). This reformulated TSP instance can then itself be solved to and then used to reconstruct the global tour. We first attempt a solution using Concorde [App+03], and the process times out (we set a limit of 1 minute) then we attempt a solution using

the LKH heuristic. We then repeat this process over all violating pairs iteratively updating the tour after each modification. It may also be possible to reconstruct the tour by doing a targeted k-opt swap on the tour at the expense of accuracy, however given that each pair is usually very small we decide to just solve the reconstruction optimally since it is relatively cheap to do so.

To evaluate the performance of this repairing heuristic, we run it on the TSPLIB [Rei91] dataset and evaluate the reduction in the optimality gap of the nearest neighbor tour after repairing the tour using the violating pairs from the WSPD. There are 76 EUC_2D problems in the dataset, and across all the s-values this gives us a total of 456 runs of the repairing heuristic.

7.3 Results

7.3.1 $S = 1.5$

We first evaluate the heuristic on problems which are guaranteed to be rearrangeable by the heuristic. These results are shown in fig. 7.1. After running over all the problems instances, we identify two: *kroA100* and *pcb3038*, for which the nearest neighbor path has a violation of the conditions for some pair in the WSPD with $s = 1.5$. For each of these problems only a single pair violates the conditions. We then apply the repair method as described in the last section and find that for each problem by rearranging the tour over the violating pair we can get a significant improvement to the tour. Specifically, for *kroA100* we get a 20.4% reduction in the optimality gap of the tour, and for *pcb3038* we get a $\approx 32.4\%$ reduction in the optimality gap of the tour.

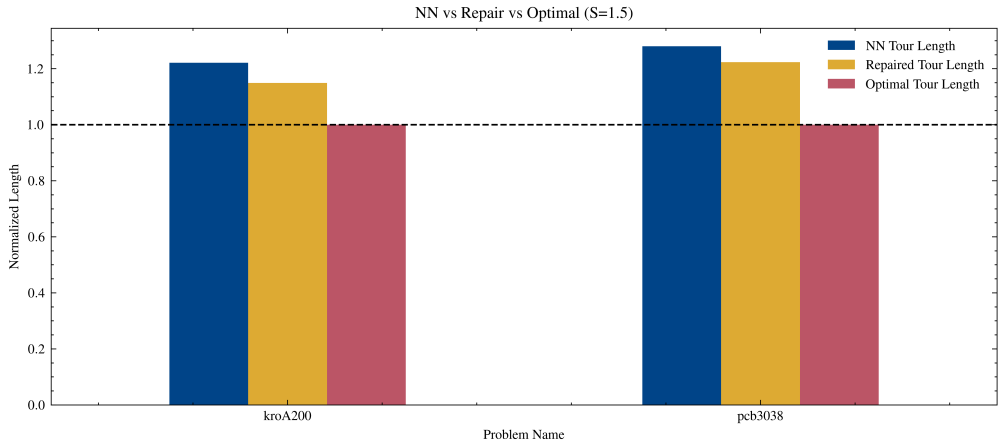
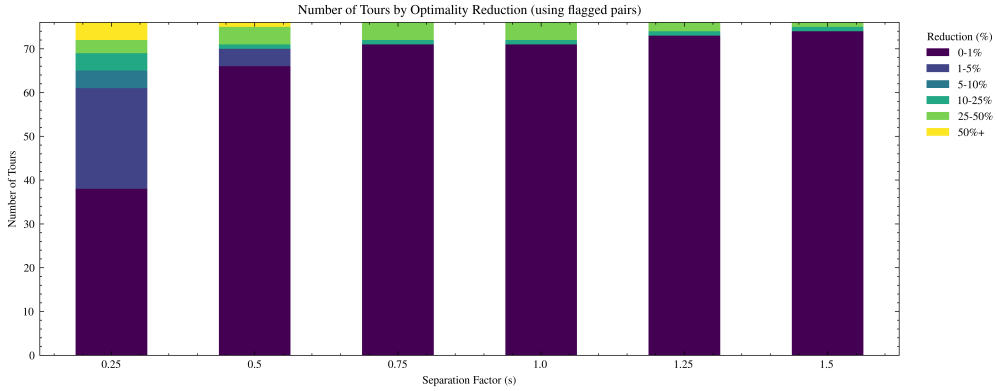


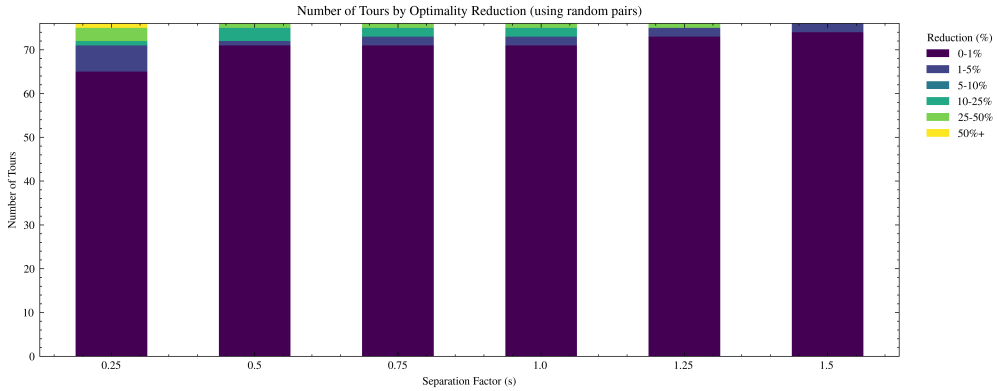
Figure 7.1: Results of the guaranteed repair heuristic for $s = 1.5$.

7.3.2 Varying Separation Factor

Next we continuously ablate the separation factor s to evaluate the performance of the heuristic as we relax the theoretical threshold. As we relax the separation factor we find that a greater number of pairs violate the conditions, and thus we have more areas of interest to modify the tour. After applying the repair method to all violating pairs for each value of s, we find that the optimality gap reduction generally increases as we relax the separation factor. The results of this experiment are shown in fig. 7.2a. Starting with the single instance reductions of 20.4% and 32.4% for $s = 1.5$, we find that by the time we get to $s = 0.25$ there are 11 problems with reductions of over 10% of the optimality gap and even 4 instances with reductions of over 50% of the optimality gap. Most impressively 50% of the problems have a reduction of over 1% showing that this setup can derive improvements to the tours over a number of problems.



(a) Repairing from violating pairs from the WSPD.



(b) Repairing from random pairs of points.

Figure 7.2: Count of tours with optimality gap reductions by separation factor s.

7.3.3 Ablation Study

To evaluate the importance of the WSPD, we run an ablation study where instead of repairing around the violating pairs from the WSPD, we randomly select nodes and run the repairs over those. Specifically we modify our original repair code by stopping at the point where we would have repaired the tour around the violating pairs, and instead we utilize a random shuffling across all nodes to get a repair section over the same number of nodes but that may now come from anywhere in the problem. This is illustrated in fig. 7.2 where we show that the number of repaired tours with a certain reduction in the optimality gap between the nearest neighbor tour and the

optimal tour is much higher when repairing around the violating pairs from the WSPD than the randomized method. These results can be further seen in fig. 7.3 where we see that only two cases have a better reduction under the random repair method than the WSPD repair method versus 92 cases where our method is better.

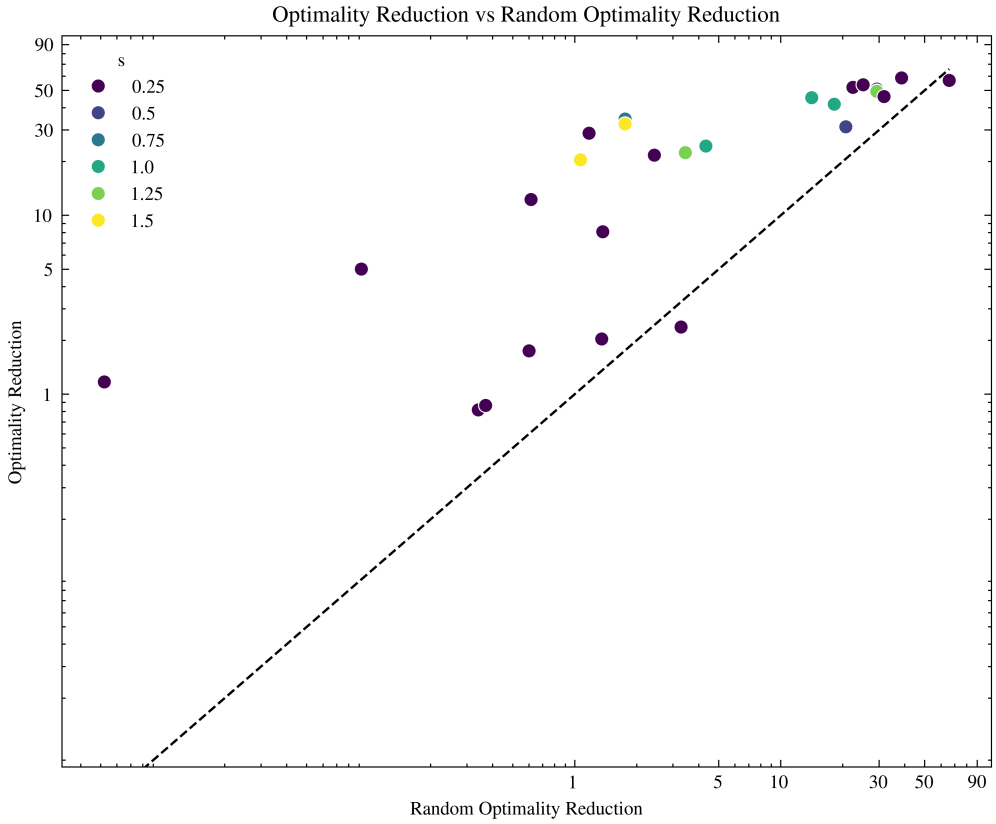


Figure 7.3: Scatter plot of each tour comparing flagged pair repair methods vs randomized pair repair methods.

Chapter 8: Conclusion

8.1 Related Work

8.1.1 Well-Separated Pair Decomposition

The well-separated pair decomposition was first introduced by Callahan and Kosaraju [CK95] as a tool for solving the k-nearest neighbor problem and the n-body problem. Since then, it has been used to solve numerous problems, including the construction of t-spanners [FG10], the problem of finding MSTs [NZ02], diameter approximation [Har25], all nearest neighbors [GSW22], and many more [SS10; GZ05; LWZ16; SSA09; Smi18].

Following Callahan and Kosaraju's seminal work, various methods for constructing WSPDs have been proposed. Some specialize in specific metric spaces [Cha08; Mat23; GZ05; Gov+06; PM13; GW25; Der+25; HM05], while others target general metric spaces such as the work of Har-Peled, Raichel, and Robson [HRR25] and Buchin et al. [Buc+26].

Although effective in many domains, these algorithms often prioritize size and speed over other factors critical to us, as detailed in chapter 5. Specifically, many existing approaches lack the generalization to all metric spaces as required for general ΔTSP instances. Furthermore, some of our results here require that the WSPD have a global structure that restricts top-level pairs to two sets or some other small number, which cannot be guaranteed by standard algorithms. Finally, achieving computational speedups through our findings requires well-balanced pairs, an objective

typically overlooked by traditional WSPD construction methods.

8.1.2 Edge Elimination

Our results from chapters 3 and 4 can be viewed as edge eliminations of the same style as Hougardy and Schroeder [HS14] and Jonker and Volgenant [JV84]. Depending on the selected theorem our proofs can be thought of as showing that certain groupings of edges between the well-separated sets are k-incompatible for some k and thus can be eliminated from consideration in an optimal TSP tour. In the case of Corollary 3.1.2.1 this is equivalent to showing that any $k > 2$ edges between the sets are k-incompatible. In the case of Theorem 3.5.3 this is equivalent to show that any edge between the sets is *useless* (not occurring in any optimal TSP tour). While our results are similar in style to those of Hougardy and Schroeder, they differ in two key ways. Firstly, our theorems expressly require the existence of sufficently well-separated sets and therefore a metric space whereas Hougardy and Schroeder's can be applied to any symmetric TSP instance. Secondly, our algorithms we build upon our theorems may be applied to any metric TSP problem, whereas Hougardy and Schroeder's algorithms are designed for the EUC_2D TSP. The same largely applies to the work of Jonker and Volgenant however when compared their results are not as effective as those of Hougardy and Schroeder on randomly generated EUC_2D problems [Zho26].

8.1.3 Clustered TSP

The Clustered TSP (CTSP) is a variant of the TSP where the points are partitioned into clusters and the goal is to find a tour that visits all points in each cluster before moving to the next cluster, while minimizing the total distance traveled [Chi75; JV85]. A 5/3-approximation

algorithm is known for the Ordered-CTSP (where the order of the clusters is fixed), which was developed by Anily, Bramel, and Hertz [ABH99]. From Corollaries 3.1.2.1 and 4.1.1.1, it follows that when given two or three well-separated clusters, there exists a $5/3$-approximation algorithm for the TSP over those clusters.

Many works come close to our results; however, our contributions vary in several ways. The earliest is that of Karp [Kar77], who describes a probabilistic algorithm for partitioning a set of points into groups and then solving the CTSP over those points by optimally solving a subpath over each group and then patching the groups together. Their work is fundamentally different from ours in that theirs lacks any theoretical guarantees, whereas we have proven our methods up to certain conditions. Furthermore, our work is designed to function for any metric space, whereas their work is designed for the Euclidean plane.

A similar study along this line is that of Xu and Lan [XL23] who utilizes k-means clustering to partition points into clusters, solves the CTSP over those clusters, and finally applies heuristics such as 2-opt to improve upon the final path. By using k-means the authors are able to better distribute their clusters in hub-and-spoke style problems where a constant number of cities per cluster would otherwise complicate the process. To account for irregular point distributions, Agostinelli [Ago17] uses density-based spatial clustering to partition points as opposed to k-means.

Again, our work differs in that we seek to determine which problems are guaranteed to have a certain structure in their optimal TSP solution and then leverage that structure to solve the TSP more efficiently, whereas their work is designed as a approximation algorithm that seeks to find good solutions to the TSP more quickly.

8.1.4 ΔTSP Heuristics

The most popular heuristics for solving the ΔTSP are based on utilizing the MST of the points and the triangle inequality. Originally this was done using the double-tree algorithm which achieves a 2-approximation. This was improved upon by Christofides [Chr22] whose algorithm achieves a 1.5-approximation by first finding the MST of the points, then finding a minimum weight perfect matching over the odd-degree vertices of the MST, and finally patching the MST and the perfect matching to form a tour. Most recently this was improved upon by Karlin, Klein, and Gharan [KKG21] whose algorithm achieves a $(1.5-\epsilon)$-approximation. By avoiding the worst-case scenarios of Christofides, they demonstrated that the approximation ratio of 1.5 for ΔTSP is not tight.

8.2 Future Work

Our work suggests a number of avenues for future work.

Min-Max Multi-TSP While we have shown a number of theoretical results for the multiple TSP through sections 3.2 to 3.4, these theorems may not be strongly applicable to real-world problems. Consider Theorem 3.4.4, which states that for $s \geq 1$ some optimal solution to the problem is to have a single path visit all points in B consecutively; while this may be mathematically optimal, it unfairly burdens one path over the others. In many cases, we may prefer a more balanced solution where the paths are more evenly distributed in terms of their lengths. This leads to the problem of the min-max multi-TSP, where the goal is to find k paths that cover all points while minimizing the maximum length of any path. It would be interesting to determine if our results can be extended to

this problem and to again identify the number of crossings in the optimal solution.

Theoretical Proofs for $s < 1$ In section 6.2, we show that for problems with s-well-separated bipartitions for $s < 1$, it is often the case that only two edges cross the bipartition in an optimal TSP solution. However, we do not have theoretical results that guarantee this property for any specific value of $s < 1$. It would be interesting to investigate whether, combined with some other problem properties, a guarantee can be established for cases where $s < 1$. We analyze multiple WSP cases where we require each set to be pairwise well-separated for $s \geq 1$; however, relaxing this requirement for certain pairs may yield further insights.

Integration with TSP Solvers While we have developed the ability to apply our results to approximated tours to find better tours—and to break down certain tours into smaller sub-problems—these techniques are not yet integrated into state-of-the-art TSP solvers such as Concorde or LKH. Our results could significantly speed up these solvers; by combining our structural insights with existing heuristics, solvers could prune suboptimal paths more efficiently and focus on promising candidates earlier. Strong results have been shown for similar techniques already by Cook et al. [Coo+24].

Applications to Metric Subsections and Near Metric TSP Currently, our results are focused on the ΔTSP; however, practical applications often involve non-metric TSP problems, including those with asymmetric distances or violations of the triangle inequality. It would be valuable to investigate how our results might apply when specific sub-regions of a problem are known to be metric, potentially allowing for the elimination of non-optimal tours. Another promising direction involves extending our findings to near-metric TSP problems, where the triangle inequality is only

slightly violated or distances are marginally asymmetric. Since many applications—such as those involving real-world road networks—are near-metric, adapting our structural results to these cases could significantly broaden their utility.

Improving Approximation Algorithms We also consider how WSPDs could be used to improve approximation algorithms for the ΔTSP. Can we provably improve upon the current best-known bound of $1.5 - \epsilon$ for all variants of the ΔTSP by utilizing WSPDs to find superior tours? If not, can we significantly improve the bound in cases where the WSPD follows a specific structure, such as having a small number of pairs or especially well-balanced pairs?

Applications to Semi-separated Pair Decomposition A slightly relaxed version of the WSPD which has been recently gaining traction is the semi-separated pair decomposition (SSPD), introduced by Varadarajan [Var98], which requires that only one of the sets have a small diameter relative to the separation distance. After its conception, the SSPD has been used to solve a number of problems, including geometric t-spanners [Aba+07; Ali+09; AH12], closest-pair queries [Aba+13], map matching queries [GSW24b], and more [GSW24a]. Given the similarities between the WSPD and SSPD, it would be interesting to investigate whether our results can be extended to the SSPD as well. If so, this would significantly broaden the applicability of our results, as the requirements of the SSPD are less strict than those of the WSPD, making it more likely that real-world problems will satisfy them.

TSP with Time Windows The Traveling Salesman Problem with Time Windows (TSPTW) is a variant of the TSP where each point must be visited within a specific time window [Sav85; Bak83; Dum+95]. While we have shown the benefits of using WSPDs to breakdown the spatial nature of

ΔTSP, we wonder whether such methods may also be applied to the temporal aspect of TSPTW. This may be feasible by encoding the time windows as an additional dimension in the metric space or by applying a separate analysis to the time windows themselves and then interleaving this with our spatial analysis.

Bibliography

[Aba+07] M. A. Abam et al. “Region-Fault Tolerant Geometric Spanners.” In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA ’07. USA: Society for Industrial and Applied Mathematics, Jan. 7, 2007, pp. 1–10. ISBN: 978-0-89871-624-5. URL: https://dl.acm.org/doi/10.5555/1283383.1283384 (visited on 03/26/2026).

[Aba+13] Mohammad Ali Abam et al. “On the Power of the Semi-Separated Pair Decomposition.” In: *Computational Geometry* 46.6 (Aug. 1, 2013), pp. 631–639. ISSN: 0925-7721. DOI: 10.1016/j.comgeo.2013.02.003. URL: https://www.sciencedirect.com/science/article/pii/S092577211300014X (visited on 03/27/2026).

[ABH99] Shoshana Anily, Julien Bramel, and Alain Hertz. “A -Approximation Algorithm for the Clustered Traveling Salesman Tour and Path Problems.” In: *Operations Research Letters* 24.1–2 (Feb. 1999), pp. 29–35. ISSN: 01676377. DOI: 10.1016/S0167-6377(98)00046-7. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167637798000467 (visited on 03/10/2026).

[Ago17] Matthew Agostinelli. “Density-Based Clustering Heuristics for the Traveling Salesman Problem.” In: *Open Access Master’s Theses* (Jan. 1, 2017). DOI: 10.23860/thesis-agostinelli-matthew-2017. URL: https://digitalcommons.uri.edu/theses/1073.

[AH12] Mohammad A. Abam and Sariel Har-Peled. “New Constructions of SSPDs and Their Applications.” In: *Computational Geometry*. Special Issue: 26th Annual Symposium on Computation Geometry at Snowbird, Utah, USA 45.5 (July 1, 2012), pp. 200–214. ISSN: 0925-7721. DOI: 10.1016/j.comgeo.2011.12.003. URL: https://www.sciencedirect.com/science/article/pii/S0925772111000976 (visited on 03/27/2026).

[Ali+09] Mohammad Ali Abam et al. “Geometric Spanners for Weighted Point Sets.” In: *Algorithms - ESA 2009*. Ed. by Amos Fiat and Peter Sanders. Berlin, Heidelberg: Springer, 2009, pp. 190–202. ISBN: 978-3-642-04128-0. DOI: 10.1007/978-3-642-04128-0_17.

[App+03] David Applegate et al. *Concorde : A Code for Solving Traveling Salesman Problems*. Dec. 19, 2003. URL: https://www.math.uwaterloo.ca/tsp/concorde.html.

[App+06] David L. Applegate et al. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006. ISBN: 978-0-691-12993-8. JSTOR: j.ctt7s8xg. URL: https://www.jstor.org/stable/j.ctt7s8xg (visited on 04/07/2026).

[Aro98] Sanjeev Arora. "Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and Other Geometric Problems." In: *J. ACM* 45.5 (Sept. 1, 1998), pp. 753–782. ISSN: 0004-5411. DOI: 10.1145/290179.290180. URL: https://dl.acm.org/doi/10.1145/290179.290180 (visited on 03/04/2026).

[Bak83] Edward K. Baker. "An Exact Algorithm for the Time-Constrained Traveling Salesman Problem." In: *Operations Research* 31.5 (1983), pp. 938–945. ISSN: 0030-364X. JSTOR: 170895. URL: https://www.jstor.org/stable/170895 (visited on 04/28/2026).

[Bel62] Richard Bellman. "Dynamic Programming Treatment of the Travelling Salesman Problem." In: *J. ACM* 9.1 (Jan. 1, 1962), pp. 61–63. ISSN: 0004-5411. DOI: 10.1145/321105.321111. URL: https://dl.acm.org/doi/10.1145/321105.321111 (visited on 04/07/2026).

[BŞ13] Hüsamettin Bayram and Ramazan Şahin. "A New Simulated Annealing Approach for Travelling Salesman Problem." In: *Mathematical and Computational Applications* 18.3 (Dec. 2013), pp. 313–322. ISSN: 2297-8747. DOI: 10.3390/mca18030313. URL: https://www.mdpi.com/2297-8747/18/3/313 (visited on 04/06/2026).

[Buc+26] Kevin Buchin et al. *On Small Pair Decompositions for Point Sets*. Feb. 2, 2026. DOI: 10.48550/arXiv.2601.22728. arXiv: 2601.22728 [cs]. URL: http://arxiv.org/abs/2601.22728 (visited on 03/27/2026). Pre-published.

[Cha08] Timothy M. Chan. "Well-Separated Pair Decomposition in Linear Time?" In: *Information Processing Letters* 107.5 (Aug. 16, 2008), pp. 138–141. ISSN: 0020-0190. DOI: 10.1016/j.ipl.2008.02.008. URL: https://www.sciencedirect.com/science/article/pii/S0020019008000550 (visited on 12/30/2025).

[Chi75] James A. Chisman. "The Clustered Traveling Salesman Problem." In: *Computers & Operations Research* 2.2 (Sept. 1, 1975), pp. 115–119. ISSN: 0305-0548. DOI: 10.1016/0305-0548(75)90015-5. URL: https://www.sciencedirect.com/science/article/pii/0305054875900155 (visited on 03/10/2026).

[Chr22] Nicos Christofides. "Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem." In: *Operations Research Forum* 3.1 (Mar. 3, 2022), p. 20. ISSN: 2662-2556. DOI: 10.1007/s43069-021-00101-z. URL: https://doi.org/10.1007/s43069-021-00101-z (visited on 12/30/2025).

[CK95] Paul B. Callahan and S. Rao Kosaraju. "A Decomposition of Multidimensional Point Sets with Applications to *k* -Nearest-Neighbors and *n* -Body Potential Fields." In: *Journal of the ACM* 42.1 (Jan. 3, 1995), pp. 67–90. ISSN: 0004-5411, 1557-735X. DOI: 10.1145/200836.200853. URL: https://dl.acm.org/doi/10.1145/200836.200853 (visited on 12/30/2025).

[Coo+24] William Cook et al. "Local Elimination in the Traveling Salesman Problem." In: *Mathematical Programming Computation* 16.4 (Dec. 1, 2024), pp. 599–628. ISSN: 1867-2957. DOI: 10.1007/s12532-024-00262-y. URL: https://doi.org/10.1007/s12532-024-00262-y (visited on 04/07/2026).

[Der+25] Lindsey Deryckere et al. *A WSPD, Separator and Small Tree Cover for c-Packed Graphs*. May 11, 2025. DOI: 10.48550/arXiv.2505.06884. arXiv: 2505.06884 [cs]. URL: http://arxiv.org/abs/2505.06884 (visited on 03/27/2026). Pre-published.

[DFJ54] G. Dantzig, R. Fulkerson, and S. Johnson. "Solution of a Large-Scale Traveling-Salesman Problem." In: *Journal of the Operations Research Society of America* 2.4 (1954), pp. 393–410. ISSN: 0096-3984. JSTOR: 166695. URL: https://www.jstor.org/stable/166695 (visited on 04/17/2026).

[dSFA21] Roberto da Silva, Eliseu Venites Filho, and Alexandre Alves. "A Thorough Study of the Performance of Simulated Annealing in the Traveling Salesman Problem under Correlated and Long Tailed Spatial Scenarios." In: *Physica A: Statistical Mechanics and its Applications* 577 (Sept. 1, 2021), p. 126067. ISSN: 0378-4371. DOI: 10.1016/j.physa.2021.126067. URL: https://www.sciencedirect.com/science/article/pii/S037843712100340X (visited on 04/06/2026).

[Dum+95] Yvan Dumas et al. "An Optimal Algorithm for the Traveling Salesman Problem with Time Windows." In: *Operations Research* 43 (Apr. 1, 1995), pp. 367–371. DOI: 10.1287/opre.43.2.367.

[FG10] Mohammad Farshi and Joachim Gudmundsson. "Experimental Study of Geometric T-Spanners." In: *ACM J. Exp. Algorithmics* 14 (Jan. 5, 2010), 3:1.3–3:1.39. ISSN: 1084-6654. DOI: 10.1145/1498698.1564499. URL: https://dl.acm.org/doi/10.1145/1498698.1564499 (visited on 03/10/2026).

[Gov+06] Sathish Govindarajan et al. "I/O-Efficient Well-Separated Pair Decomposition and Applications." In: *Algorithmica* 45 (Aug. 1, 2006), pp. 585–614. DOI: 10.1007/s00453-005-1197-3.

[GR69] J. C. Gower and G. J. S. Ross. "Minimum Spanning Trees and Single Linkage Cluster Analysis." In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 18.1 (1969), pp. 54–64. ISSN: 0035-9254. DOI: 10.2307/2346439. JSTOR: 2346439. URL: https://www.jstor.org/stable/2346439 (visited on 02/07/2026).

[GSW22] Anirban Ghosh, F. N. U. Shariful, and David Wisnosky. *Visualizing WSPDs and Their Applications*. May 9, 2022. DOI: 10.48550/arXiv.2205.04308. arXiv: 2205.04308 [cs]. URL: http://arxiv.org/abs/2205.04308 (visited on 03/10/2026). Pre-published.

[GSW24a] Joachim Gudmundsson, Martin P. Seybold, and Sampson Wong. *Approximating Multiplicatively Weighted Voronoi Diagrams: Efficient Construction with Linear Size*. Mar. 18, 2024. DOI: 10.48550/arXiv.2112.12350. arXiv: 2112.12350 [cs]. URL: http://arxiv.org/abs/2112.12350 (visited on 03/27/2026). Pre-published.

[GSW24b] Joachim Gudmundsson, Martin P. Seybold, and Sampson Wong. "Map Matching Queries on Realistic Input Graphs Under the Fréchet Distance." In: *ACM Trans. Algorithms* 20.2 (Mar. 13, 2024), 14:1–14:33. ISSN: 1549-6325. DOI: 10.1145/3643683. URL: https://dl.acm.org/doi/10.1145/3643683 (visited on 03/27/2026).

[GW25] Joachim Gudmundsson and Sampson Wong. *A Well-Separated Pair Decomposition for Low Density Graphs*. Oct. 13, 2025. DOI: 10.48550/arXiv.2411.08204. arXiv: 2411.08204 [cs]. URL: http://arxiv.org/abs/2411.08204 (visited on 03/27/2026). Pre-published.

[GZ05] Jie Gao and Li Zhang. "Well-Separated Pair Decomposition for the Unit-Disk Graph Metric and Its Applications." In: *SIAM Journal on Computing* 35.1 (Jan. 2005), pp. 151–169. ISSN: 0097-5397, 1095-7111. DOI: 10.1137/S0097539703436357. URL: http://epubs.siam.org/doi/10.1137/S0097539703436357 (visited on 01/02/2026).

[Har25] Sariel Har-Peled. *A Practical Approach for Computing the Diameter of a Point Set*. May 16, 2025. DOI: 10.48550/arXiv.2505.11317. arXiv: 2505.11317 [cs]. URL: http://arxiv.org/abs/2505.11317 (visited on 03/10/2026). Pre-published.

[Hel09] Keld Helsgaun. "General K-Opt Submoves for the Lin–Kernighan TSP Heuristic." In: *Mathematical Programming Computation* 1.2 (Oct. 1, 2009), pp. 119–163. ISSN: 1867-2957. DOI: 10.1007/s12532-009-0004-6. URL: https://doi.org/10.1007/s12532-009-0004-6 (visited on 04/06/2026).

[HHH24] Sophia Heimann, Hung P. Hoang, and Stefan Hougardy. "The K-Opt Algorithm for the Traveling Salesman Problem Has Exponential Running Time for k $\geq$ 5." In: *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*. Ed. by Karl Bringmann et al. Vol. 297. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 84:1–84:18. ISBN: 978-3-95977-322-5. DOI: 10.4230/LIPIcs.ICALP.2024.84. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2024.84 (visited on 04/06/2026).

[HK61] Michael Held and Richard M. Karp. "A Dynamic Programming Approach to Sequencing Problems." In: *Proceedings of the 1961 16th ACM National Meeting*. ACM '61. New York, NY, USA: Association for Computing Machinery, Jan. 1, 1961, pp. 71.201–71.204. ISBN: 978-1-4503-7388-3. DOI: 10.1145/800029.808532. URL: https://dl.acm.org/doi/10.1145/800029.808532 (visited on 04/07/2026).

[HM05] Sariel Har-Peled and Manor Mendel. "Fast Construction of Nets in Low Dimensional Metrics, and Their Applications." In: *Proceedings of the Twenty-First Annual Symposium on Computational Geometry*. SCG '05. New York, NY, USA: Association for Computing Machinery, June 6, 2005, pp. 150–158. ISBN: 978-1-58113-991-4. DOI: 10.1145/1064092.1064117. URL: https://dl.acm.org/doi/10.1145/1064092.1064117 (visited on 03/27/2026).

[HRR25] Sariel Har-Peled, Benjamin Raichel, and Eliot W. Robson. *Well-Separated Pairs Decomposition Revisited*. Sept. 7, 2025. DOI: 10.48550/arXiv.2509.05997. arXiv: 2509.05997 [cs]. URL: http://arxiv.org/abs/2509.05997 (visited on 01/02/2026). Pre-published.

[HS14] Stefan Hougardy and Rasmus T. Schroeder. "Edge Elimination in TSP Instances." In: *Graph-Theoretic Concepts in Computer Science*. Ed. by Dieter Kratsch and Ioan Todinca. Cham: Springer International Publishing, 2014, pp. 275–286. ISBN: 978-3-319-12340-0. DOI: 10.1007/978-3-319-12340-0_23.

[HW15] Stefan Hougardy and Mirko Wilde. "On the Nearest Neighbor Rule for the Metric Traveling Salesman Problem." In: *Discrete Applied Mathematics*. 12th Cologne-Twente Workshop on Graphs and Combinatorial Optimization (CTW 2013) 195 (Nov. 20, 2015), pp. 101–103. ISSN: 0166-218X. DOI: 10.1016/j.dam.2014.03.012. URL: https://www.sciencedirect.com/science/article/pii/S0166218X14001486 (visited on 04/06/2026).

[JK91] Chang-Sung Jeong and Myung-Ho Kim. "Fast Parallel Simulated Annealing for Traveling Salesman Problem on SIMD Machines with Linear Interconnections." In: *Parallel Computing* 17.2 (June 1, 1991), pp. 221–228. ISSN: 0167-8191. DOI: 10.1016/S0167-8191(05)80107-3. URL: https://www.sciencedirect.com/science/article/pii/S0167819105801073 (visited on 04/06/2026).

[JV84] Roy Jonker and Ton Volgenant. "Nonoptimal Edges for the Symmetric Traveling Salesman Problem." In: *Operations Research* 32.4 (1984), pp. 837–846. ISSN: 0030-364X. JSTOR: 170580. URL: https://www.jstor.org/stable/170580 (visited on 04/07/2026).

[JV85] Kees Jongens and Ton Volgenant. "The Symmetric Clustered Traveling Salesman Problem." In: *European Journal of Operational Research* 19.1 (Jan. 1, 1985), pp. 68–75. ISSN: 0377-2217. DOI: 10.1016/0377-2217(85)90309-1. URL: https://www.sciencedirect.com/science/article/pii/0377221785903091 (visited on 03/10/2026).

[Kar77] Richard M. Karp. "Probabilistic Analysis of Partitioning Algorithms for the Traveling-Salesman Problem in the Plane." In: *Mathematics of Operations Research* 2.3 (1977), pp. 209–224. ISSN: 0364-765X. JSTOR: 3689510. URL: https://www.jstor.org/stable/3689510 (visited on 03/10/2026).

[KKG21] Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. "A (Slightly) Improved Approximation Algorithm for Metric TSP." In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2021. New York, NY, USA: Association for Computing Machinery, June 15, 2021, pp. 32–45. ISBN: 978-1-4503-8053-9. DOI: 10.1145/3406325.3451009. URL: https://dl.acm.org/doi/10.1145/3406325.3451009 (visited on 03/04/2026).

[Kre89] M. W. Krentel. "Structure in Locally Optimal Solutions." In: *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*. SFCS '89. USA: IEEE Computer Society, Oct. 30, 1989, pp. 216–221. ISBN: 978-0-8186-1982-3. DOI: 10.

1109/SFCS.1989.63481. URL: https://doi.org/10.1109/SFCS.1989.63481 (visited on 04/06/2026).

[LK73] S. Lin and B. W. Kernighan. "An Effective Heuristic Algorithm for the Traveling-Salesman Problem." In: *Operations Research* 21.2 (1973), pp. 498–516. JSTOR: 169020. URL: http://www.jstor.org/stable/169020.

[LWZ16] Fabian Lipp, Alexander Wolff, and Johannes Zink. "Faster Force-Directed Graph Drawing with the Well-Separated Pair Decomposition." In: *Algorithms* 9.3 (Sept. 2016), p. 53. ISSN: 1999-4893. DOI: 10.3390/a9030053. URL: https://www.mdpi.com/1999-4893/9/3/53 (visited on 03/27/2026).

[Mat23] Domagoj Matijević. "Well-Separated Pair Decompositions for High-Dimensional Datasets." In: *Algorithms* 16.5 (May 2023), p. 254. ISSN: 1999-4893. DOI: 10.3390/a16050254. URL: https://www.mdpi.com/1999-4893/16/5/254 (visited on 03/27/2026).

[Mit96] Joseph S. B. Mitchell. "Guillotine Subdivisions Approximate Polygonal Subdivisions: A Simple New Method for the Geometric k-MST Problem." In: *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '96. USA: Society for Industrial and Applied Mathematics, Jan. 28, 1996, pp. 402–408. ISBN: 978-0-89871-366-4. URL: https://dl.acm.org/doi/10.5555/313852.314090 (visited on 04/21/2026).

[Mur+23] V. Murugananthan et al. "Traveling Salesman Problem with Ant Colony Optimization." In: *2023 2nd International Conference on Edge Computing and Applications (ICECAA)*. 2023 2nd International Conference on Edge Computing and Applications (ICECAA). July 2023, pp. 481–485. DOI: 10.1109/ICECAA58104.2023.10212262. URL: https://ieeexplore.ieee.org/document/10212262 (visited on 04/06/2026).

[NZ02] Giri Narasimhan and Martin Zachariasen. "Geometric Minimum Spanning Trees via Well-Separated Pair Decompositions." In: *ACM J. Exp. Algorithmics* 6 (Dec. 31, 2002), 6–es. ISSN: 1084-6654. DOI: 10.1145/945394.945400. URL: https://dl.acm.org/doi/10.1145/945394.945400 (visited on 03/10/2026).

[PM13] Eunhui Park and David M. Mount. "Output-Sensitive Well-Separated Pair Decompositions for Dynamic Point Sets." In: *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPATIAL'13. New York, NY, USA: Association for Computing Machinery, Nov. 5, 2013, pp. 354–363. ISBN: 978-1-4503-2521-9. DOI: 10.1145/2525314.2525364. URL: https://dl.acm.org/doi/10.1145/2525314.2525364 (visited on 03/10/2026).

[PY93] Christos H. Papadimitriou and Mihalis Yannakakis. "The Traveling Salesman Problem with Distances One and Two." In: *Mathematics of Operations Research* 18.1 (1993), pp. 1–11. ISSN: 0364-765X. JSTOR: 3690150. URL: https://www.jstor.org/stable/3690150 (visited on 03/04/2026).

[Rei91] Gerhard Reinelt. “TSPLIB—A Traveling Salesman Problem Library.” In: *ORSA Journal on Computing* 3.4 (Nov. 1991), pp. 376–384. ISSN: 0899-1499. DOI: 10.1287/ijoc.3.4.376. URL: https://pubsonline.informs.org/doi/10.1287/ijoc.3.4.376 (visited on 03/10/2026).

[Sav85] M. W. P. Savelsbergh. “Local Search in Routing Problems with Time Windows.” In: *Annals of Operations Research* 4.1 (Dec. 1, 1985), pp. 285–305. ISSN: 1572-9338. DOI: 10.1007/BF02022044. URL: https://doi.org/10.1007/BF02022044 (visited on 04/28/2026).

[Sch18] Jan Scholz. “Genetic Algorithms and the Traveling Salesman Problem a Historical Review.” 2018. DOI: 10.13140/RG.2.2.22632.78088/1. arXiv: 1901.05737 [cs]. URL: http://arxiv.org/abs/1901.05737 (visited on 04/06/2026).

[Smi18] Michiel Smid. “The Well-Separated Pair Decomposition and Its Applications.” In: *Handbook of Approximation Algorithms and Metaheuristics*. 2nd ed. Chapman and Hall/CRC, 2018.

[SS10] Jagan Sankaranarayanan and Hanan Samet. “Query Processing Using Distance Oracles for Spatial Networks.” In: *IEEE Transactions on Knowledge and Data Engineering* 22.8 (Aug. 2010), pp. 1158–1175. ISSN: 1558-2191. DOI: 10.1109/TKDE.2010.75. URL: https://ieeexplore.ieee.org/document/5494757 (visited on 03/27/2026).

[SSA09] Jagan Sankaranarayanan, Hanan Samet, and Houman Alborzi. “Path Oracles for Spatial Networks.” In: *Proc. VLDB Endow.* 2.1 (Aug. 1, 2009), pp. 1210–1221. ISSN: 2150-8097. DOI: 10.14778/1687627.1687763. URL: https://dl.acm.org/doi/10.14778/1687627.1687763 (visited on 04/21/2026).

[Tal04] Kunal Talwar. “Bypassing the Embedding: Algorithms for Low Dimensional Metrics.” In: *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*. STOC ’04. New York, NY, USA: Association for Computing Machinery, June 13, 2004, pp. 281–290. ISBN: 978-1-58113-852-8. DOI: 10.1145/1007352.1007399. URL: https://dl.acm.org/doi/10.1145/1007352.1007399 (visited on 03/04/2026).

[THW18] Renato Tinós, Keld Helsgaun, and Darrell Whitley. “Efficient Recombination in the Lin-Kernighan-Helsgaun Traveling Salesman Heuristic.” In: *Parallel Problem Solving from Nature – PPSN XV*. Ed. by Anne Auger et al. Cham: Springer International Publishing, 2018, pp. 95–107. ISBN: 978-3-319-99253-2. DOI: 10.1007/978-3-319-99253-2_8.

[Tsa+14] Chun-Wei Tsai et al. “A High-Performance Genetic Algorithm: Using Traveling Salesman Problem as a Case.” In: *The Scientific World Journal* 2014 (2014), p. 178621. ISSN: 2356-6140. DOI: 10.1155/2014/178621. PMID: 24892038. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC4032739/ (visited on 04/06/2026).

[Var98] K.R. Varadarajan. "A Divide-and-Conquer Algorithm for Min-Cost Perfect Matching in the Plane." In: *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*. 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280). Nov. 1998, pp. 320–329. DOI: 10.1109/SFCS.1998.743466. URL: https://ieeexplore.ieee.org/document/743466 (visited on 03/27/2026).

[XL23] Haiyang Xu and Hengyou Lan. "An Adaptive Layered Clustering Framework with Improved Genetic Algorithm for Solving Large-Scale Traveling Salesman Problems." In: *Electronics* 12.7 (Jan. 2023), p. 1681. ISSN: 2079-9292. DOI: 10.3390/electronics12071 URL: https://www.mdpi.com/2079-9292/12/7/1681 (visited on 03/10/2026).

[Zho26] Xianghui Zhong. "Probabilistic Analysis of Edge Elimination for Euclidean TSP." In: *Mathematics of Operations Research* 51.1 (Jan. 2026), pp. 621–640. ISSN: 0364-765X. DOI: 10.1287/moor.2020.0299. URL: https://pubsonline.informs.org/doi/abs/10.1287/moor.2020.0299 (visited on 04/07/2026).