ABSTRACT

Title of dissertation: APPROXIMATION ALGORITHMS FOR GEOMETRIC CLUSTERING AND TOURING PROBLEMS

Ioana Oriana Bercea
Doctor of Philosophy, 2018

Dissertation directed by: Professor Samir Khuller
Department of Computer Science

Clustering and touring are two fundamental topics in optimization that have been studied extensively and have "launched a thousand ships". In this thesis, we study variants of these problems for Euclidean instances, in which clusters often correspond to sensors that are required to cover, measure or localize targets and tours need to visit locations for the purpose of item delivery or data collection.

In the first part of the thesis, we focus on the task of sensor placement for environments in which localization is a necessity and in which its quality depends on the relative angle between the target and the pair of sensors observing it. We formulate a new coverage constraint that bounds this angle and consider the problem of placing a small number of sensors that satisfy it in addition to classical ones such as proximity and line-of-sight visibility. We present a general framework that chooses a small number of sensors and approximates the coverage constraint to arbitrary precision.

In the second part of the thesis, we consider the task of collecting data from a

set of sensors by getting close to them. This corresponds to a well-known generalization of the Traveling Salesman Problem (TSP) called TSP with Neighborhoods, in which we want to compute a shortest tour that visits at least one point from each unit disk centered at a sensor. One approach is based on an observation that relates the optimal solution with the optimal TSP on the sensors. We show that the associated bound can be improved unless we are in certain exceptional circumstances for which we can get better algorithms.

Finally, we discuss Maximum Scatter TSP, which asks for a tour that maximizes the length of the shortest edge. While the Euclidean version admits an efficient approximation scheme and the problem is known to be NP-hard in three dimensions or higher, the question of getting a polynomial time algorithm for two dimensions remains open. To this end, we develop a general technique for the case of points concentrated around the boundary of a circle that we believe can be extended to more general cases.

APPROXIMATION ALGORITHMS FOR GEOMETRIC CLUSTERING AND TOURING PROBLEMS

by

Ioana Oriana Bercea

Dissertation 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
Doctor of Philosophy
2018

Advisory Committee:
Professor Samir Khuller, Chair/Advisor
Professor John Yiannis Aloimonos
Professor William Gasarch
Professor David Mount
Professor Mark Shayman

All culture is like this: the unfettered mind,

The boundless spirit's mere imagination,

For pure perfection's heights will strive in vain.

To achieve great things, we must be self-confined:

Mastery is revealed in limitation

And law alone can set us free again.

—Johann Wolfgang von Goethe, *Nature and Art*

Acknowledgments

I gratefully acknowledge Prof. Samir Khuller for being a strong and wonderful advisor. Under his guidance, I have grown to be a better researcher and a more inquisitive scientist. I have learned that curiosity and courage are powerful allies when thinking about problems or optimizing life. I am proud to have been his student while he was chair of the department, and to work with him as he was also showing us how to be a leader. Through personal example, Samir has set for the rest of us the high standard of always being able to explain complicated techniques in intuitive and elegant terms. I can only say that I will continue to strive to reach it. I am grateful for his perpetual wisdom and unwavering support and will be returning for his guidance in the years to come.

I also count in my academic family the collaborators I have had throughout the years. I would like to thank Dr. Navin Goyal for welcoming me to Microsoft Research Bangalore as a young Phd student, and for encouraging me to pursue my ideas confidently. I am grateful for Prof. Volkan Isler's support and for his enthusiasm for theoretical approaches. Chapter 3 is the fruit of our attempt to bridge the gap between practice and theory, and in that sense, visiting his lab at the University of Minnesota has taught me to have great respect towards the practitioners of our field. Prof. Parinya Chalermsook and Prof. Mayank Goswami hosted me at the Max Planck Institute for Informatics in Saarbrücken and showed me how to aim high and tackle the hardest of problems. Working with Prof. Aravind Srinivasan and Prof. David Harris at UMD has been a masterclass. I have also enjoyed working with my

labmates Saba Ahmadi, Aounon Kumar and Sheng Yang and thank them for the lively group meetings. My work has many connections to Prof. Joe Mitchell's work and I would like to thank him for the deep conversations we have had throughout the years. I would especially like to thank him and Tyler Mayer for hosting me at Stony Brook University and for creating an amazing research environment. Finally, I would like to thank Prof. Tobias Mömke for his true intellectual camaraderie and all the joyful conversations. Chapter 5 contains the beginning of our work together, done while visiting MPI.

My committee members are some of my favorite professors in the CS department. I consider myself lucky to have had a chance to learn from them and I thank them for the interest they took in my doctorate. From Prof. David Mount, I have learned the art of making clean arguments while being joyful and truly passionate about the work. I will never forget all the insightful conversations I had with Prof. Yiannis Aloimonos, especially about Euclid's *Elements* and the philosophical geometry of the world. Seeing Prof. William Gasarch teach and being his TA has taught me how to encourage young talent to evolve and how to be a maverick in my thinking.

My extended UMD family includes many dear friends, and I thank them all for their friendship and care. Special love to my brothers and sisters in arms Ejaz Ahmed, Martha Beard, Rajesh Chitnis, Sameh Khamis, Ben London, Moschoula Pternea, Thodoris Rekatsinas, Octavian Suciu, Aishwarya Thiruvengadam and Kostas Zampogiannis. Manish Purohit and Kanthi Sarpatwar welcomed me into Samir's group and I am proud to call them my academic brothers. In that vein, I would

also like to thank Prof. Rajiv Gandhi for his kindness and support when I started working with Samir. Lastly, I would like to acknowledge the lovely people that supported me during stressful administrative adventures and who always brought their warmth, Adelaide Findlay, Jodie Gray, Sharron McElroy, Stephanie Peters and Jennifer Story.

Outside of UMD, I would like to thank the Algorithms and Complexity group at MPI for making me feel welcomed and for giving me two great summers. I would like to specifically thank Prof. Kurt Melhorn for giving me the opportunity to visit his wonderful group. Guy Even, Moti Medina and Emanuele Natale have made my time there unforgettable and I would like to thank them for their constant encouragement and inspiration.

My life would not have been the same without the influence of my first family outside of Romania: the University of Chicago. My alma mater has given me the unbelievable opportunity to pursue my potential and all of my intellectual dreams. In recent days, I have come to revisit some old favorites from those undergraduate years. My admission letter and the statement of full financial support still bring tears to my eyes. My essays from that time remind me of the true Socratic tradition in which I was immersed and of the friendships that have endured since then. Andreea Balica, Natalia Hryniw, Kim-Mei Kirtland, Brian Libgober and Brian Mayer have been wonderful companions in the life of the mind and I thank them for their support.

I am also grateful for the great education given to me by Prof. László Babai, who introduced me to Theoretical Computer Science via the rigorous and elegant

proof techniques from Combinatorics. Prof. Diane Herrmann has had a lasting effect on my life and her care and attention will always make the UChicago Mathematics Department feel like home. I would also like to thank the Romanian school of Mathematics for providing an inspiring environment in which to train and learn the discipline of Mathematics and the joy of proving theorems. Whenever the circles look particularly divine, I am reminded of the poetry of Mathematics just as I experienced it growing up.

Closer to my heart, I would like to thank two people that are very important to me. The extraordinary Daniel M. Choi has been the Thelma to my Louise, and every fight that I have fought has been his as well. In our search for Ithaca, we will forever travel. Erik Jan van Leeuwen has been "bold, inventive and fearless. That's who you wanna go in the woods with, right? Somebody who finishes your sentences for you" (Tom Waits).

Most importantly, I would like to thank my talented parents, who have been my most important champions. Their wisdom, courage and passion guide and nourish me always. They are my roots and I always carry them in my core wherever I go. Words fail to describe how grateful I feel so I can only say "Vă pup şi vă iubesc!"

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Of all optimization paradigms in our toolkit, casting problems as instances of clustering or touring is perhaps one of the most popular and widespread approaches. Consequently, attacking problems in these frameworks has spawned a multitude of techniques. In the face of ever-changing scenarios coming from the applied fields, the renewed challenge is to develop methods that accommodate for more complicated constraints or objectives. A theoretical approach to these issues often times brings unexpected insight into how we can obtain good approximate solutions. The goal then becomes that of bringing fresh theoretical insights while keeping in mind the original motivation and eventual applicability of our techniques.

This thesis was born out of the desire to understand how we can exploit the geometry of the input space in designing better approximation algorithms and a story about tracking fish. Specifically, the problems of clustering and touring appear naturally in the context of the carp monitoring framework introduced by Bhadauria et al [2].

The Common Carp is a highly invasive fish that once present in a lake, unbalances the entire ecosystem through its feeding pattern. In the Minnesota lakes, significant effort has been invested in controlling the spread of this non native fish. One established approach to controlling such populations has been to study the social biology of the species. In this direction, one important observation was made by Bajer et al [3]: adult carp tend to aggregate in certain regions of the lake for long periods of time (for example, in feeding or breeding areas). When identified as such, large numbers of fish can be efficiently removed through netting.

In order to detect such groups, Bajer et al [3] employ the Judas technique, which is based on the idea that the social nature of some invasive species can be used to betray their location [4]. Specifically, the biologists surgically implant some of the fish with radio transmitters and release them back into the lake. The fish are then tracked while they relocate and join the groups they usually live with. This involves releasing boats into the lake while scientists explore various locations (known and unknown) and try to accurately locate the Judas fish according to the radio signals they emit.

A major challenge in this complex task of active monitoring is that it can be expensive and daunting in terms of human resources. In this context, Bhadauria et al [2] propose a model that automates this task by employing light weight robotic rafts. The goal is to use a small number of rafts to correctly identify the Judas fish, potentially using some knowledge about the most likely places where it could be. In subsequent work, Tokekar et al [5] propose a variety of algorithms for this and identify two distinct stages: the search phase, in which the boat scans areas of

the lake where the fish is likely to be present, and the localization phase, in which, once a tagged fish is detected, several bearing measurements are taken in a way that minimizes uncertainty.

The first phase therefore roughly corresponds to computing a shortest tour that visits targets contained in a polygonal space (i.e. the lake) in a Traveling Salesman fashion. Moreover, in order for the boat to sense the presence of a fish in one of these regions, it generally needs to be within a specified range of it (100-200m). Once a fish is detected, several measurements are needed to localize it, so in the second phase, the boat needs to choose sensing locations from which it can compute the location of the tagged fish to arbitrary precision.

Inspired by this framework, the first problem that this thesis considers is that of selecting a small number of stationary sensors that can accurately localize targets. We cast this as a clustering problem and focus on ensuring that each possible target location is appropriately covered. This would roughly correspond to having a guess on where the fish locations might be and wanting to always be able to monitor those locations accurately.

In the second part of this thesis, we consider the mobile aspect of active monitoring and compute short tours that visit regions of interest. We do this by furthering our structural understanding of specific touring problems. As a consequence, we develop more efficient algorithms and gain insight that we believe can have practical value.

1.1 Context and Applications

We begin by describing the landscape in which our problems lie and provide some history behind them. We also discuss some practical applications for our techniques and mention existing general approaches.

1.1.1 Clustering with Uncertainty

Clustering is one of the fundamental problems in Computer Science, with applications in optimization, machine learning and operations research [6, 7]. It essentially centers around grouping input elements together such that the groups satisfy certain local criteria and the clustering itself optimizes a global measure. Depending on what the underlying formulation is, we can roughly classify them into two categories:

- problems in which we want to select the smallest number of clusters such that the membership of each element to a cluster satisfies a specific set of constraints,
- problems in which we are given a fixed budget to spend on these clusters and we want to cover the elements such that we optimize a global objective.

Together, these two directions have given birth to some of the most famous problems in Theoretical Computer Science, such as Set Cover and Facility Location respectively. Since their formulation, they have been adapted to accommodate a plethora of optimization scenarios and rich theories have been developed

around them. To this day, they continue to remain relevant and adapt to current concerns such as privacy and fairness [8, 9]. In Chapter 2, we review some of the relevant techniques and problems.

Coverage constraints. In this thesis, we focus mainly on the first category of problems and specifically the scenario in which each cluster corresponds to a sensor and the task at hand is that of observing targets represented as input points. In general, sensor placement problems are formulated as *coverage* problems in which each element needs to be covered by a sensor. Given a set of target locations (potentially infinite) and the underlying geometry, a coverage constraint is defined that depends on the problem at hand. The corresponding optimization problem becomes that of placing a small number of sensors that achieve the desired coverage.

For example, in the *Art Gallery problem* as defined by O'Rourke [10], the targets lie inside a polygon P and the coverage constraint is that of line of sight visibility: a guard (sensor) at location s sees a target at location t if the segment $\overline{st}$ is contained inside the polygon P. This constraint is relevant especially in the context of urban environments that might exhibit occlusions.

In open environments, the problem can be that of covering the polygon P with sensors that have a limited sensing range and are therefore modeled as disks, also known as the *Disk Packing/Placement* problem. More sophisticated coverage functions, encountered for example in 3D reconstruction, can address the directionality of a sensor (such as a camera) or its orientation with respect to the surface that contains t. See the survey of Scott et al [11] for more details on such scenarios. In essence, coverage is classically defined as a boolean function that depends on one

sensor covering one target.

Covering for localization. A more complex notion of coverage, however, involves not just detecting a target, but also taking measurements on it for the purpose of localizing or describing it. Localization is an important necessity in many mobile computing applications. In ad-hoc wireless sensor networks, it centers around the ability of nodes to self localize using little to no absolute spatial information. When a large number of sensors are deployed, it becomes impractical to equip all of them with the capability of localizing themselves with respect to a global system (such as through GPS). On the other hand, while GPS can be used for localization in outdoor settings, localization in indoor environments remains a challenge [12]. In parallel, as "smart" home, warehouse and factory automation applications gain traction, providing location services in such settings is becoming more important. When mobility is considered, the problem becomes that of tracking a moving target through a sensor network in which a set of sensors must combine measurements in order to detect the location of the target.

In this context, most problem formulations rely on the assumption that for each location, measurements from a single sensor are sufficient (such as in temperature or light measurements). There are important situations, however, in which this is not the case. For example, because of their limited capacity, most sensors cannot directly localize a target. Cameras can only measure the relative bearing[1] of t and

[1]In navigational terms, relative bearing corresponds to the angle between the current sensor's forward direction and the location of the target. For example, an object right in front of the camera would have a relative bearing of 0. An object right to the left of the camera would have a

as such, their measurement corresponds to a ray that originates from s and passes through t (insufficient information if we want to pinpoint exactly the location of t). From this perspective, a commonly used technique for localization is to employ a small number anchors (or beacons) that know their location and are capable of transmitting it to the other nodes seeking to localize themselves [13]. Alternatively, sensors such as cameras or microphone arrays placed in the environment can collect measurements which can then be used to estimate the locations of objects of interest [14,15].

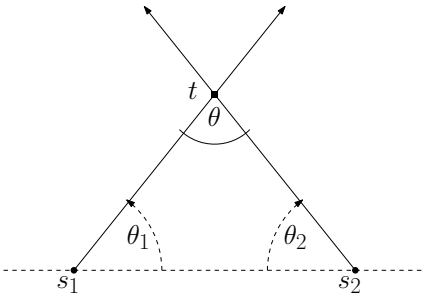


Figure 1.1: The two fixed observers are s_1 and s_2. The target to be localized is sitting at position t. The measurements available are the relative bearings $\theta_1 = \angle ts_1s_2$, $\theta_2 = \angle ts_2s_1$ and the distance d between s_1 and s_2. Such measurements determine uniquely the position of t, as it is the intersection of two rays starting at s_1 and s_2 respectively and passing through t.

In both approaches, some of the most popular measurements used are Euclidean distance and angle between pairs of nodes (bearing), such as is the case with Received Signal Strength Indicator (RSSI), time-to-arrival (ToA) or Angle-of-Arrival (AoA). In this context, each target seeking to localize itself has access to Euclidean distances and/or angular measurements relative to the sensors that are in its vicinity. When exact distances or bearings from two sensors to a target are known, localization can be easily performed through the process of triangulation, as seen in Figure 1.1 and discussed by Williams [16].

Benefits of multiple coverage. Another reason why assigning multiple sensors to a target is a beneficial strategy is that of *energy minimization*. Given

relative bearing of $\pi/2$.

that sensors have limited energy, it can make sense to deploy a group of sensors to observe the same target but have only one sensor per group active at any given time. As shown by Slijepcevic et al [17], activating the sensors from each group in a round robin fashion can lead to an extended lifespan of the entire network. On the other hand, multiple sensors that achieve the same task ensure the property of *fault-tolerance*: when one of the sensors fails, the other ones can act as backup. Often times, these properties have been considered as variants of several clustering and graph theoretic problems as discussed by Abrams et al [18], Khuller et al [19,20] and Kuhn et al [21].

Tackling uncertainty. An even more fundamental problem is that there is an inherent *uncertainty* in each of the measurements taken. In this context, new models have been introduced that require measurements from at least two distinct sensor locations in order to get a reasonable estimate. For example, some formulations require measurements from multiple sensors to jointly converge on the location of a target, as in the case of the sculpture garden problem of Eppstein, Goodrich and Sitchinava [22]. Others define visibility to depend on the convex hull of the sensors, like in the case of the triangle guarding problem considered by Smith and Evans [23].

Another way of dealing with error estimation introduced by Cressie [24] has been to learn a probability distribution of uncertainty over the entire field to be observed and to place sensors where the uncertainty is the highest. The problem with this approach is that while focusing on measuring locations of high uncertainty, it overlooks the quality of the measurement on the overall space. A more recent result of Kraus et al [25] deals directly with the efficiency of such measurements

and incorporates it into the uncertainty objective to be minimized. Alternatively, models have been considered that compute Cramer-Rao bounds as lower bounds for the accuracy of calibration [26–29].

1.1.2 Touring Problems

Deployment of sensors, for example, is often only the first step in such problem formulations arising in practice. A variety of other scenarios arise when we allow these sensors to be mobile or we have a robot with motor capabilities. Broadly speaking, when the positions of targets are roughly known, the task becomes that of computing the shortest path along which the robot can travel, collect data from sensors in the field and then return to its original starting position. This is, at its core, the well known Traveling Salesperson Problem (TSP), in which we would like to compute a shortest tour that visits a given set of locations.

TSP is perhaps one of the most notoriously hard problems in Theoretical Computer Science, in the sense in which getting better approximation algorithms for it has resisted attack throughout time [30, 31]. For example, in general graphs that respect the triangle inequality, we have the foundational result of Christofides [32] that gives a 1.5-approximation. Breaking this barrier remains an outstanding open problem as the true constant is conjectured to be 4/3 [33–35]. Progress has been made in the special case in which the underlying graph is geometric, where we have the celebrated polynomial-time approximation schemes(PTAS) of Arora [36], later derandomized by Rao and Smith [37], and Mitchell [38]. We also mention

the recent significant progress on another long standing open question regarding the *asymmetric* version of TSP. Specifically, Svensson et al [39] have shown the first constant factor approximation for the case in which the underlying graph is directed and the distance function is non-negative, satisfies triangle inequality but might not be symmetric. For example, this models instances in which edges represent streets whose cost is different depending on which direction we traverse them: they might be one-way, congested with one-directional traffic or uphill.

Geometric Context. The success of the Euclidean case has give rise to a number of efficient algorithms for a variety of other touring problems. We refer the reader to the survey of Kumar et al [40]. We mention some here as an opportunity to give a succinct view of the breadth of such problems:

- *minimum latency problems*, also know as the *Traveling Repairman Problem* in which we want to minimize the time each customer has to wait before the traveling repairman serves them [41, 42],
- *orienteering problems*, in which we have an upper bound on the length of the tour and we want to visit as many sites as possible [43],
- *capacitated vehicle routing problems* in which we have access to multiple trucks, each capable of visiting k sites and we want to minimize the total length of paths [44].

Moreover, Euclidean TSP has proven essential in a variety of applications and a lot of heuristics have been proposed for it. On one hand, we have formulations that depend on the capabilities of the robot/truck, such as in the *Angle-Restricted*

Tour introduced by Fekete et al [45]. In this formulation, given set of allowable angles, we ask whether a certain point set admits a tour in which each turn has an angle that is from the allowed set (i.e. only right 90° turns, left 90° turns etc). This accounts for the fact that, in practice, rotation is very limited and/or more expensive than translation.

Similarly, the *Angular-Metric TSP* problem introduced by Aggarwal et al [46] requires that the total sum of direction changes in the tour is minimized (i.e. the direction change ranges from 0, when the next edge in the tour lies on the same line as the one before it, and 180°, when the next edge turns back around on the same line). The authors also mention that an interesting open problem would be to require that the direction change at each turn be bounded and the length of the tour minimized: this is, in essence, the α-coverage constraint applied at each vertex of the tour.

Finally, some machines require that the distance they travel between successive points be large: this is formulated as the *Maximum Scatter TSP* problem which asks for the tour that maximizes the length of the shortest edge. The specific motivation behind this problem comes from the manufacturing industries where a certain task has to be performed in different locations (such as high precision laser cutting on a metal sheet) and each location experiences a certain regional secondary effect once visited (such as the temperature of the surrounding area increasing). In these circumstances, we want to avoid visiting locations that are too close to each other in succession because we want to avoid provoking a large secondary effect. In the laser cutting context, this would amount to a particular area exhibiting higher

temperatures than the rest, which could negatively effect the whole outcome. Some approaches deal with this by consequently imposing that successive locations on the tour be well separated [47]. This is indeed the Maximum Scatter TSP problem, which we discuss in Chapter 5 in more detail.

On the other hand, some applications depend on more complex decision making which must adapt a shortest tour to fit within a certain overarching task. For example, in the *Data Gathering Problem* defined by Tekdas et al [48], the sensors are stationary while several robots are used to travel between them in order to collect data. Additional costs such as the time it takes to gather the data make this problem considerably different from TSP because now, multiple robots can go along the same long path and gather data faster. This task of data collection in wireless sensor networks has been garnering a lot of attention recently both in the theory and in more applied circumstances [49–51].

Another exciting direction comes from the practical application of home delivery of goods [52]. Specifically, recent work has been focused on understanding the benefit of utilizing drone technology that can support deliveries. This is known as *TSP with drone* [53] or *Vehicle Routing Problem with Drones* [54]. In this formulation, we have a truck that is equipped with several drones such that the drones are faster than the truck but have a limited battery life or storage capacity. The challenge then becomes to decide at which points along the tour should the truck release the drones such that the overall delivery time is sped up. In particular, this takes advantage of the inherent parallelism of having multiple drones that can accomplish several fast deliveries simultaneously. Indeed, recent work has shown

this leads to significant speedup in delivery times [54,55] and various heuristics have been proposed [53].

TSPN. Finally, one of the most well studied generalizations of TSP is the *Traveling Salesman Problem with Neighborhoods* (TSPN). In this problem, the notion of visiting points is relaxed to that of visiting regions. Specifically, a region is considered to be visited if the tour intersects it at any point contained in it. Our Chapter 4 will be concerned with approximation algorithms for the case in which each region is modeled as a uniform disk. We will defer the technical results for there but we mention here that this problem is not just a natural realization of TSP on points but also arises naturally in practice. For instance, it is also known as the *Close Enough TSP* and has been applied to the task of automated meter reading from a distance [56]. The main observation is that due to modern advances in radio technology, a utility company only need to get within range of a certain customer in order to read their utility consumption. We are then tasked with finding a shortest tour that must visit the customer within some range, rather than traveling to the customer directly. Similarly, in the Data Gathering Problem defined above, each sensor has a limited communication range R and hence, the mobile robot must visit disks of radius R centered at the sensors in order to receive the data. Other variations use multiple robots [50, 57] or consider the tradeoff between R and the time it takes to download the data have also been considered [58]. When it comes to using drone technology, one can envision already formed clusters of points that need only be visited at a cluster representative (and have the drones released there): this would correspond to the discrete version of TSPN, also known as *Group TSP* or

One-of-a-Kind TSP [59]. All of these formulations have as their basis the TSPN on uniform disks problem.

A lot of heuristics have been proposed for this problem. One of the most natural approaches is to select a set of points to visit and then return a TSP tour on those points [60]. For example, one could cluster the points first such that each of them is within some distance of the cluster head and then visit only the cluster heads [61]. Alternatively, one could identify zones of overlap that, if visited, would satisfy a lot of demands [62]. The construction of the tour can start with an intuitive guess (such as a convex hull) and can then be improved using simulated annealing [63], rubber band or artificial bee colony algorithms [64]. Evolutionary [49, 65] or mixed integer programming techniques [56] can also be used. Yuan et al. [49] provide heuristics that fix the order in which to visit the disks and then cast the problem of deciding where each disk gets visited as a continuous optimization problem. Carrabs et al. [60] consider various discretizations of the boundary of each disk and then solve a linear integer program to decide in which order to visit the disks. Most of these works implement their algorithms on instances of ≈ 100 points and test the length of the solutions produced and the runtime of the algorithm against established benchmarks.

In Chapter 4, we discuss the TSPN problem on unit disks in more detail.

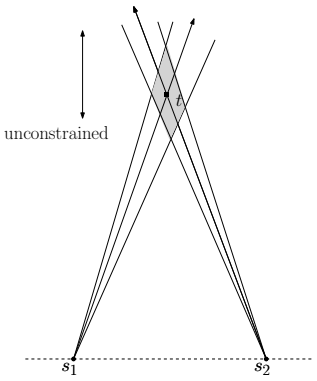
1.2 Motivation and Contributions

We are now ready to introduce the specific problems we study and the particular motivation behind them. We identify the main questions that we strive to answer and briefly review our results.

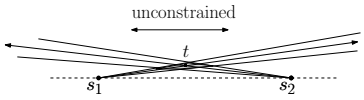
1.2.1 Clustering with Angular Constrains

We begin by casting uncertainty as a function of the geometry that underlies the localization task. Specifically, from a geometric perspective, the target/sensor geometry plays a significant role in the quality of location estimates. In this context, another common benchmark for localization performance is the Geometric Dilution of Precision (GDOP) that investigates how the geometry between the sensors and the target nodes amplifies measurement errors and affects the localization error. Savvides et al. [28, 29] observe that the error is largest when the angle θ between two sensors and the target node is either very small or close to π.

The analysis of Kelly [66] further shows that, when triangulation is used, this angle contributes to the GDOP at a fundamental level. When distance measurements are used in triangulation, the GDOP is proportional to $1/|\sin\theta|$. When angular measurements are used, the GDOP is proportional to $d_1 \cdot d_2/|\sin\theta|$, where d_1 and d_2 are the distances from the sensors to the node. In general, distance information comes from connectivity of the communication graph and depends on the sensing range of the sensor. As such, it can be modeled as a disk or annulus. Bearing



(a) When the angle θ subtended by the target and the two sensors is small, the product of ranges is high.



(b) As θ get close to π, localization becomes impossible.

Figure 1.2: Notice that in both cases, the shape of the intersection degenerates in one of the directions (x or y) and the feasible set of target locations becomes unconstrained.

information is subject to additive error and can be modeled as a cone. The above measurements become constraints on the possible location of the target, each restricting the set of feasible locations. Intuitively, the quality of localization depends on both the area and the shape of the intersection of the feasible regions defined by the measurements. When the angle θ is close to 0 or π, the intersection becomes unconstrained and the error unbounded. In particular, when the sensors and the target are collinear, localization is impossible. In essence, overall uncertainty is minimized when the sensors are well separated angularly about the target.

Inspired by these observations, our work focuses on the geometry of sensor deployment and asks the question of **where should the sensors be placed such that we control the inherent uncertainty in measurement at the GDOP level**. Specifically, given a set of candidate sensor locations and a set of possible target locations, what is the minimum number and placement of sensors so as to ensure

that the uncertainty in estimating the target's location is below a given threshold for all possible target locations? The question then becomes one of coverage and, to this end, we define an angular constraint which we call α-*coverage*: given a parameter $\alpha \in [0, \pi/2]$, each target at position t must be assigned two sensors (at positions s_1 and s_2) such that the angle $\theta = \angle s_1 t s_2$ is in the range $[\alpha, \pi - \alpha]$ (i.e. neither too small nor too big). Bounding the angle in this fashion allows us to upper bound its influence on the GDOP.

We then frame the problem of sensor placement as a bicriteria optimization problem: given a set of possible sensor and target locations, we wish to select the smallest number of sensors that provide α-coverage for all target locations. This formulation captures the scenario in which we have discretized the space into finitely many locations. We study this problem for a number of settings in conjunction with other classical constraints such as sensing range and line-of-sight visibility. We address these variants from a theoretical perspective and present a general algorithmic framework that specifically addresses the angular constraint and iteratively obtains better angular guarantees at the expense of larger solution sizes.

This work appeared in the proceedings of the 28th Canadian Conference on Computational Geometry [67].

1.2.2 The Traveling Salesman Problem with Neighborhoods

The first challenge we encounter in designing better algorithms for TSPN comes from the fundamental way in which it differs from the TSP: the solution

requires us to not only decide in which order to visit the regions but also which points to actually visit in each region. In this context, the first constant factor algorithms proposed by Dumitrescu and Mitchell [68] are based on the idea of picking a specific point from each of the disks and returning an approximate TSP on these points as a solution. In the disjoint case, they consider an approximate TSP tour on the *centers* of each disk and show that it gives a good constant factor approximation to the original problem.

The core of their analysis is a bound that compares the length of the optimal TSP on the centers of each disk ($|TSP^*|$) with the length of the optimal TSPN on the disks ($|TSPN^*|$) and says that

$$|TSP^*| \leq |TSPN^*| + 2Rn, \tag{1.1}$$

where $n \geq 2$ is the number of disks in the instance. In addition, the authors use a packing argument to lower bound the length of the optimal TSPN tour in terms of R and n and get that $\frac{\pi}{4}Rn - \pi R \leq |TSPN^*|$. Overall, this gives a 3.547-approximation and in addition, the authors show that the algorithm cannot give better than a factor 2 approximation. As can be seen in Figure 1.3(a), the $2Rn$ factor in (1.1) comes from a detour approach: any tour touching the boundary of the disks can be turned into a tour touching the centers of each disk by going to each center and coming back. This incurs a cost of $2R$ per disk.

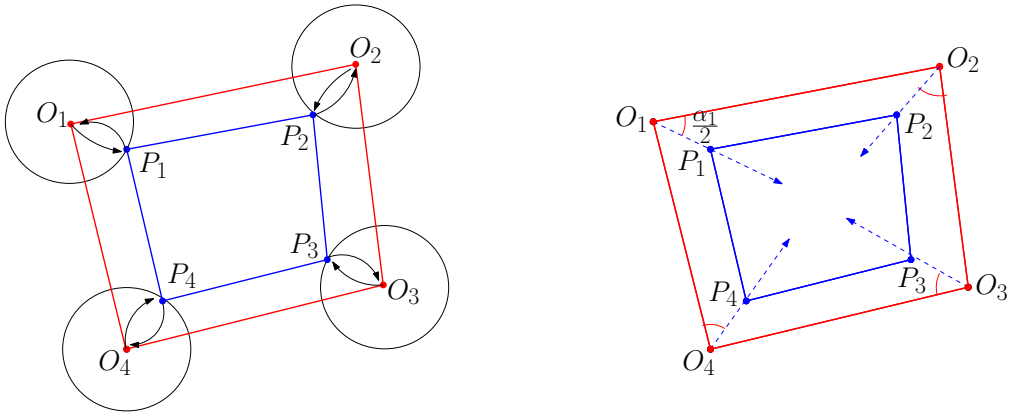
While other methods for choosing representative points can be employed [68–70], this approach is appealing both in its elegance and because it does not depend on R. In addition, the tour on the centers becomes a fundamental benchmark

against which to compare a variety of other tours, such as in TSP with drones or data gathering. Understanding the trade-off between visiting disks at their centers and visiting them anywhere on their boundary becomes important in designing more nuanced algorithms for such problems.

Moreover, other existing constant factors approximations for TSPN often hide large constants [69,71,72] that are incurred as a consequence of using general bounds on the length of the optimal tour that do not directly exploit the structure of the regions or of the optimal TSPN tour (bounding rectangle argument and Combination Lemma in [70]). In order to improve on them, the challenge then becomes to develop bounds that exploit the difference in behavior between a TSP tour (on points) and the TSPN tour on the regions and furthermore, avoid using general purpose techniques that add on to the overall approximation factor.

In this context, one way to improve the approximation factor for disjoint disks is to better understand the relationship between the optimal TSP tour on the centers and the optimal TSPN on the disks. Specifically, **is the $2Rn$ term in (1.1) tight or can it be improved by using specific structural properties of the optimal TSPN?**

A similar question was asked in 2011 by Häme, Hyytiä and Hakula [73] for the case when R is very small (and hence, TSP^* and $TSPN^*$ respect the same order and the disks are pairwise disjoint). They conjectured that the true detour term should be $\sqrt{3}nR$ and constructed arbitrarily large instances of disjoint disks that converge to this case. We refer to this as the **Häme, Hyytiä and Hakula conjecture.** Subsequent experiments by Müller [74], however, suggest that this might be true



(a) The detour approach.

(b) The differential approach.

Figure 1.3: The TSP on the centers is drawn in red and the TSPN in blue. The detour approach imagines the TSPN making a detour of at most $2R$ per disk in order to visit each center. In the differential approach, we imagine the centers moving along the bisector at uniform speed. The TSPN then corresponds to the tour when the centers have each moved a distance R.

only for tours up to five disks and higher otherwise. No further progress has been made towards the conjecture since then.

The basis of the Häme, Hyytiä and Hakula conjecture is their *differential neighborhoods* approach, in which they start by considering a tour on the centers and ask how should points be moved at most R away from the centers such that the new tour on these points has the smallest length possible (Figure 1.3(b)). They further define the *shortening rate* of a fixed order to be the ratio of the decrease in tour length $|TSP^*| - |TSPN^*|$ over R, as $R \to 0$. A bound of $2Rn$ implies a shortening rate of at most $2n$ and hence the formulations are equivalent up to the fact that, for general R, the order in which the TSP visits the centers and that in which the TSPN visits the disks might not be the same.

This differential approach has implications beyond bounding the TSPN in

terms of the TSP, as it is related to the theory of Euclidean curve shortening and the rendezvous problem for mobile autonomous robots. In the latter, the robots start at fixed positions and the goal is to develop a moving strategy for them such that they eventually converge to a common point, preferably only using local information [75]. The perspective that is relevant to us was considered by Smith et al [76] who further take into account the shape of the formation of robots as they are converging. In this sense, an ordering on the robots gives an associated polygon (tour) and one could study how this shape changes as the robots follow a specific movement pattern. For general curves, this is known as the Euclidean curve shortening flow [77]. Smith et al. [76] investigate the problem for the case of polygons. When the ordering on the robots corresponds to the TSP^* order on their initial positions, the question becomes that of how the TSP^* changes as the robots move according to a fixed direction. Interestingly, Smith et al. [76] show that if we want to minimize the perimeter of the polygon as it changes, each point should move along the bisector of the internal angle of the polygon at that vertex. When all the points move at equal speed, Häme, Hyytiä and Hakula [73] show that this strategy implies that the shortening rate is exactly $2\sum_{i=1}^{n}\cos\frac{\alpha_i}{2}$, where α_i is the internal angle of the tour considered. What this intuitively means for us is that instances in which the TSP takes many sharp turns will push our detour term closer to $2Rn$.

We further investigate this question and derive structural properties of the optimal TSPN tour to describe the cases in which the bound is smaller than $2Rn$. Specifically, we show that if the optimal TSPN tour is not a straight line, at least one of the following is guaranteed to be true: the bound is smaller than $1.999Rn$ or

the TSP on the centers is a 2-approximation. This bound of 2 is tight, as there are instances in which the TSP on the centers is exactly twice as long as the $TSPN$ tour on the disks. Our framework is based on using the optimality of the TSPN tour to identify local structures for which the detour is large and then using their geometry to derive better lower bounds on the length of the TSPN tour. This leads to an improved approximation factor of 3.53 for the problem. We further show that the Häme, Hyytiä and Hakula conjecture is true for the case of three disks and discuss the method used to obtain it.

This work is scheduled to appear in the proceedings of the 30th Canadian Conference on Computational Geometry.

1.2.3 The Maximum Scatter Traveling Salesman Problem

While Euclidean TSP is a prime example of the benefits of considering the geometry of the underlying space in algorithm design, some instances have proven to be even more surprising in that they can be NP-hard in general graphs but have polynomial time algorithms once we consider Euclidean instances. Such is that case for Maximum TSP in polyhedral metrics in finite dimensions [78] or for rectilinear metrics in the plane [79].

In fact, a more nuanced phenomenon might be at play here. The complexity of the problem might change drastically when we differentiate between two and higher dimensions: some problems could be polynomially time solvable in two dimensions and NP-hard in higher dimensions. In fact, both Maximum TSP and Maximum

Scatter TSP are known to be NP-hard in three or higher dimensions [79] while for both, there is evidence that some planar instances are easy [80,81]. One of the most exciting questions in this direction then becomes **whether such problems are indeed polynomial in the planar Euclidean case or remain just as hard as their high dimensional counterparts**.

This open question is recognized as an important one by the Computational Geometry community and has been asked several times [79,82]. The current conjecture is that it remains NP-hard for this case. When it comes to Maximum Scatter TSP, the only progress made has been to show polynomial time algorithms for points on a line, on a circle and on specific rectangular grids [80,81]. Other interesting special cases to consider would be points lying on the boundary of an arbitrary convex body, or concentrated around the circle.

In our work, we begin by providing an alternative algorithm for the case of points on a circle. As opposed to the original one by Arkin et al. [80], our algorithm uses the geometry of the input space to derive a more general technique that does not necessarily aim to show a fixed ordering on the points that leads to a good tour but rather one that can be applied in less restrictive scenarios.

The circle case is interesting to us for an additional reason: it comes up in one of the methods we develop for the Traveling Salesman Problem with Neighborhoods in Section 4.5. Specifically, the $2Rn$ bound in 1.1 can be expressed as the Maximum TSP tour on points lying on the boundary of a circle. Knowing what the Maximum TSP tour looks like on the circle helps us identify the cases in which the detour bound for TSPN is large. Without going into details, we mention here that those

points correspond to locations that the optimal TSPN tour visits on the input disks. For example, the $2Rn$ bound is achieved when the TSPN tour alternates in hitting the disks in one of two (diametrically opposite) points. If that happens, then the TSPN tour is easy to describe and the general idea is that this is the case also for bounds close to $2Rn$ (not necessarily just for $2Rn$ exactly).

One way to understand the Maximum TSP on the circle scenario is also through Max Scatter TSP. For the case of $2Rn$, the Maximum TSP tour is actually equivalent to the Maximum Scatter TSP and intuitively, this seems to be the case in general. A secondary question we could therefore ask is whether **for the case of points on a circle, is it the case that the Maximum TSP and the Maximum Scatter TSP are equivalent?** Furthermore, if we manage to show such an equivalence, our analysis for TSPN could be vastly improved, since it would allow us to make an argument locally: the points that the TSPN touches in successive rounds actually have to be well separated on the boundary. If such a thing were true, then it would actually allow us to obtain much better general bounds on TSPN.

Chapter 2

Preliminaries

At its core, the field of Theoretical Computer Science concerns itself with the art of problem solving and the craft of algorithm design. The primary objects of study are *optimization problems*, which can broadly be defined as asking for the best solution to a certain question according to a given set of criteria. For example, in the *Traveling Salesman Problem*, we want to find the shortest tour through a city that visits a given set of locations. In the *Graph Coloring* context, one could ask for a coloring of the countries on a map using the minimum number of colors such that no two neighboring countries share the same color[1]. Finally, in the *k-center problem*, we want to know where we should open k hospitals such that we minimize the farthest distance any one customer must travel to its closest hospital.

Each of these problems asks for a structured answer (a tour, a valid coloring, a set of k hospitals) and associates a cost to each such solution (*length* of the tour,

[1]The surprising answer to this question is that 4 colors always suffice. This is famously known at the *Four Color Theorem* and has a fascinating history behind it. Although I did not research this topic, I include it as a beautiful example of the unexpected depth of easy to state problems.

number of colors used, *distance* to the closest hospital). From this perspective, the best solution is one that optimizes this cost over all potential answers (*minimum* length, number of colors and distance, although maximization problems are equally common). In addition to this, we want to find such a solution *efficiently* and in an *automated* manner that can conceivably be implemented by a computer.

More specifically, an *algorithm* consists of a series of unambiguous 'basic' instructions that, given some input (specific locations, map of the countries), terminates in finite time and outputs a solution for the problem at hand. The notion of efficiency encompasses not just how good the solution is with respect to the *optimal* one but also how computationally expensive the algorithm is with respect to resources such as time and space. In that sense, the goal of algorithm design is to exploit the structure of the problem and of the input instances and develop methods that are mindful of computational limitations.

Given a specific optimization problem, the subsequent questions that can be asked are "Can this problem be solved optimally?", "How fast can a good solution be found?", "What properties of the input instance can we use to make better decisions in our algorithm?", and so on.

2.1 Foundations of Approximation Algorithms

Formally, we begin by parameterizing our problems according to the size of the input instance. At its core, this captures the description complexity of the instance and it usually boils down to the number of input points, which we denote

as $n \in \mathbb{N}$. The runtime and space usage of the algorithm are then described in terms of this parameter n, conventionally abstracting away system specific details. One of the most important measure of runtime complexity is in terms of the *worst case* performance of the algorithm. This functions as an upper bound on runtime/space (through established big-oh notation) and it is sometimes accompanied by lower bounds that capture computational limitations. While alternative measures exist (such as average case performance), we will focus on worst case analysis throughout this thesis.

When it comes to *efficient* algorithms, the broadest definition encompasses those whose runtime and space usage is a polynomial function of n. In practice, as data sets increase in size, the race to obtain faster algorithms continues. For instance, significant progress has been made in our understanding of sublinear algorithms and runtime lower bounds. Other directions of research have successfully explored what can we achieve with very limited space.

Nevertheless, the standard of obtaining polynomial time algorithms remains an important benchmark, especially since it underlies a major division in our classification of problems. On one hand, we can start by identifying the problems for which we know a polynomial time algorithm exists. Such is the case of the b-EDGE COVER we consider in Section 3.5, or of our efforts for the MAXIMUM SCATTER TRAVELING SALESMAN PROBLEM in Chapter 4. Conventionally, they are said to belong in the computational class P.

On the other hand, we can identify problems for which a potential solution can be easily checked in polynomial time. These are problems that are said to be in

NP (Nondeterministic Polynomial). For example, the question of whether a planar country map can be validly colored with four colors is easily in NP: given a potential coloring, we can check in polynomial time that no two neighboring countries are colored with the same color. The majority of optimization problems fall in this later category.

Moreover, a fundamental breakthrough was made with the Cook-Levin theorem [83] and Karp's list of 21 NP-complete problems [84]. Collectively, they established an entire class of problems that are reducible to each other modulo polynomial time transformations. In other words, if one of the problems in the class can be optimally solved in polynomial time, all the other problems in it can be as well. This has lead to the most important open question in Theoretical Computer Science, that of "P versus NP" . At it core, the question ultimately boils down to whether NP-complete problems actually do admit polynomial time algorithms. In practice, they are generally believed to be difficult to solve optimally and even thought to capture elusive human creativity [85].

In response to this, the field of *approximation algorithms* designs polynomial time solutions to NP-complete problems that are guaranteed to be within a given factor of the optimal solution. For example, if OPT denotes the size of the optimal solution to a given minimization (maximization) problem, then an α-approximation algorithm is guaranteed to produce a solution of size at most $\alpha\cdot$ OPT (at least $\alpha\cdot$OPT). Sometimes, we can optimize for multiple parameters at the same time, in which case we have *bicriteria* approximation algorithms. This is the case in Chapter 3, when we optimize both for the number of sensor used and angular converage.

In this thesis, we concern ourselves with getting tighter approximation factors, lower for minimization problems and larger for maximization ones.

In particular, we can design *polynomial-time approximation schemes* (PTAS) that are guaranteed to get a solution of size at most $(1+\epsilon)\cdot\text{OPT}$ for any $\epsilon > 0$ in time polynomial in n for every fixed ϵ. The exponent might depend on ϵ as is for example in $\mathcal{O}(n^{1/\epsilon})$. If the exponent of n does not depend on ϵ but part of the runtime is an arbitrary function of $1/\epsilon$, we have an *efficient* polynomial-time approximation scheme (EPTAS). If the runtime also depends polynomially in $1/\epsilon$, then we call them *fully polynomial-time approximation schemes* (FPTAS). This is the case for our results in Chapter 3. In Chapter 4, we concern ourselves with lowering known constant approximation factors.

Next, we formalize some general problems and discuss some canonical approximation algorithms for clustering.

2.2 Approximation Algorithms for Clustering

In Chapter 1, we mentioned two major types of clustering problems. Now, we mention two of the most important problems associated with the categories: Set Cover and k-center/ k-suppliers. Additionally, we also discussed the idea of associating multiple sensors (cluster heads) to a particular target (input point). This is generally known as the fault tolerant case. In this section, we review known results and their associated techniques as they will become relevant later in the thesis.

To begin with, we define SET COVER in the following way, where by $|S|$ we mean the cardinality of S:

Set Cover

Input : A set system $\mathcal{F}(X, \mathcal{R})$ where X is the universe of elements and $\mathcal{R} \subseteq 2^X$ is a collection of subsets of X.

Output: A subcollection $S \subseteq \mathcal{R}$ such that $X = \cup_{C \in \mathcal{R}} C$ and $|S|$ is minimized.

In other words, the SET COVER problem asks for the smallest number of subsets that collectively cover the entire universe. The canonical algorithm known for this problem is a deceivingly simple greedy strategy: at each step, choose the set that covers the highest number of uncovered elements. This leads to a $\log n$-approximation, as shown by Johnson [86], Lovász [87] and Stein [88]. Surprisingly, this is tight: Feige [89] and Lund et al [90] have shown that the $\log n$ approximation factor is essentially the best possible for any polynomial-time algorithm. In the next section, we further describe how geometry can help to obtain better solutions.

When it comes to fault-tolerance, one version was introduced by Abrams et al [18], who consider the problem of minimizing the energy lifespan of a network of sensors and formulate it as the *Set k-cover* problem. In this problem, one must compute a set of k covers. Each cover consists of sensors that cover almost all of the targets. Together, the k covers must cover the targets as many times as possible. Intuitively, this means that each target is observed by a sensor in each of the covers, and hence, if we activate only one cover in each time step, each single sensor gets activated only once every k time steps. They provide a randomized algorithm that partitions the sensors within $(1-1/e)$ of the optimum (in expectation)

and two deterministic algorithms that achieve $1/2$ and $(1 - 1/e)$ approximation factors. They also show that obtaining a performance guarantee of better than $15/16$ is NP-complete.

The reason why this does not directly apply to our case is that our choice of sensors needs to be correlated: not only does every target need to be covered twice but the sensors assigned to it have to be in a specific relative position. We further discuss this in Section 3.6, where we adapt the original greedy strategy to our problem and show that it does not lead to improved results precisely for the aforementioned reasons.

We will also consider the *dual problem* to Set Cover, known as Hitting Set. In this problem, we can imagine transforming a Set Cover instance by exchanging input points with sets in the following manner:

- each subset in the original Set Cover instance now becomes an input point in our universe,
- for each point in the original Set Cover instance, we consider the set of subsets of $\mathcal{R}$ that cover it; this will become a set in our Hitting Set instance.

Selecting the minimum number of subsets in the Set Cover instance now corresponds to selecting a minimum number of points that collectively intersect/hit every constructed set. Formally, we have:

Hitting Set

Input : A set system $\mathcal{F}(X, \mathcal{R})$ where X is the universe of elements and $\mathcal{R} \subseteq 2^X$ is a collection of subsets of X.

Output: A set of elements $H \subseteq X$ such that $X \cap C \neq \emptyset$, $\forall C \in \mathcal{R}$ and $|H|$ is minimized.

In general, Raz et al [91] have shown that it is NP-hard to approximate the HITTING SET problem to within a factor better than $\mathcal{O}(\log n)$. Even when each element of X is guaranteed to lie in at most two sets of $\mathcal{R}$, the problem remains NP-complete [92]. Notice that an algorithm for HITTING SET can be used to obtain an approximation for SET COVER and vice versa. Our framework in Chapter 3 will not use the SET COVER formulation directly but rather reduce our algorithm to finding good hitting sets.

We will now consider the final version of clustering, that in which we have a fixed budget on the number of clusters and we want to minimize the radius of each cluster. We will consider the most general version, known as the k-SUPPLIERS problem:

k-suppliers

Input : A bipartite graph $G = (U, V, E)$ in which U is the set of suppliers and V is the set of clients, a distance function $d : U \times V \to \mathbb{R}^{\geq 0}$ and a parameter $k \in \mathbb{N}$.

Output: A set $S \subseteq U$ of k suppliers such that we *minimize* the $\max_{c \in V} \min_{s \in S} d(c, s)$.

In other words, we need to select a set of suppliers such that we minimize the

maximum distance from any point in V to its closest supplier in S. If the optimal solution to the problem is R^*, we know that every client can be serviced by a supplier within distance at most R^*. When $U = V$, this is known as the k-CENTER problem, perhaps a more well known variant.

In general metric spaces, Hochbaum and Shmoys [93] have shown a 3-approximation for k-SUPPLIERS . In the case of k-CENTER, we have better 2-approximations [93,94]. These were shown to be tight by Hsu and Nemhauser [95] unless P=NP. The asymmetric [96–98] and non-uniform cases [99] have also been considered. The wide applicability of this optimization problem has given rise to numerous variants that strive to incorporate useful constraints into the solution, such as capacity [100–102], lower bounds on the size of each cluster [103, 104] or outliers [105, 106].

In general, techniques for k-CENTER provide approximation guarantees by first guessing the optimal cluster radius and considering the associated threshold graph, a technique introduced by Hochbaum and Shmoys [93]. They obtain solutions that either cluster everything using k clusters or provide a witness that the guess radius is too low. By performing binary search on all possible cluster radii, they eventually arrive at a good guess.

Given a guess radius, one can show a lower bound on the optimal solution size. The main observation is that if two clients are far apart, they must be assigned to different suppliers. Specifically, the size of any maximal set of such far apart clients is a good lower bound for the size of the optimal set of suppliers. In other words, if we pick one supplier for each such far away client, we are guaranteed to never pick more suppliers than the optimal solution would. The algorithm then proceeds

to focus on such far apart clients and assigns suppliers to each of them optimally. The rest of the input clients must be, by definition, close to one of the far apart clients and get assigned to whatever supplier is closest to the latter. The cost of focusing on these far apart clients is that the rest of the input clients have to now travel farther than what they would have had to travel in the optimal solution. This approach gives the best approximation possible unless P=NP.

In the fault tolerant setting, Khuller et al [19] consider the *Fault Tolerant k-center* problem. In the metric fault tolerant version, we require that each point has multiple such close centers. This problem is NP-hard, since it is a generalization of k-CENTER. By the same reasoning, it is hard to approximate it within a factor better than 2 unless P=NP. Khuller et al [19] consider several variations of this problem. When we require that each point (regardless of it being a center or not) has δ other centers close to it, they provide a 3-approximation for any δ and a 2-approximation for $\delta < 4$. When they require that only the points that have not been chosen as centers have δ centers close, they provide a 2-approximation for any δ. For the *fault tolerant k-suppliers* problem, Khuller et al [19] give a 3-approximation, which is best unless P=NP.

2.3 The Computational Geometry Context

When it comes to Euclidean instances, it is generally believed that most NP-complete optimization problems become easier to approximate and can even admit polynomial time algorithms. As we have seen, the power of graph algorithms comes

from abstracting away most of the details of the input and getting to the essence of what makes a problem difficult. This leads to very general techniques with wide applicability.

Nevertheless, a significant fraction of real world problems has the additional feature that the underlying data lies in a (potentially high dimensional) Euclidean plane. In that case, it becomes worthwhile to consider this aspect as part of our problem formulation. This is the case especially since in such instances, we can often design faster data structures or develop stronger structural theorems that are not true in general.

To begin with, one of the most natural scenarios in which hitting sets are encountered is when the universe of elements consists of points in Euclidean space and the sets are induced by geometric objects. Even then, however, the problem remains NP-hard for the simple case in which the sets are induced by unit disks, as shown by Hochbaum et al [107]. Nevertheless, better than $\log n$- approximation are still possible, however.

In the special case in which the set system corresponds to certain geometric objects, a general technique introduced by Haussler and Welzl [108] is that of using ϵ-nets. Intuitively, for any $0 < \epsilon \leq 1$, an ϵ-net is a set of elements $N \subseteq X$ that intersects all the "heavy" sets of $\mathcal{R}$. In the uniform case, we require N to intersect any set $C \in \mathcal{R}$ with $|C| \geq \epsilon|X|$. More generally, consider a weight function $s\mu : X \to \mathbb{R}_{\geq 0}$ such that the weight of a subset of X is the total weight of the points in that subset. A set $A_t \in \mathcal{R}$ is called ϵ-heavy if $\mu(A_t) \geq \epsilon\mu(X)$. An ϵ-net N must then intersect all ϵ-heavy sets in $\mathcal{R}$.

Given an algorithm that computes ϵ-nets in the weighted case, the challenge is then to compute a weight function that guarantees that the ϵ-net will also be a hitting set for all the sets, not just the heavy ones. The well known algorithm of Brönnimann and Goodrich [109] iteratively computes such a set of weights and guarantees that, eventually, a $\frac{1}{2\tau}$-net will be a hitting set, where τ is the optimal hitting set. Specifically, given an algorithm that computes in polynomial time an ϵ-net of size $\mathcal{O}((1/\epsilon) \cdot g(1/\epsilon))$, they design an algorithm that returns a hitting set of size $\mathcal{O}(\tau \cdot g(\tau))$, when g is a monotonically increasing sublinear function [109].

The algorithm of Even et al [110], on the other hand, solves the fractional hitting set LP and employs the solution in building a set of weights that guarantee that an $\frac{1}{\tau^*}$-net is a hitting set, where τ^* is the value of the LP solution. The size of the hitting set would therefore be $\mathcal{O}(\tau^* \cdot g(\tau^*))$. Given an ϵ-net finder that takes polynomial time, the running times of the two algorithms are comparable and are polynomial in m and n, where $n = |\mathcal{R}|$ and $m = |X|$. In addition, in a recent result by Agarwal and Pan [111], faster algorithms are presented, that can be applied to a variety of range spaces.

Whichever algorithm we employ, the question eventually boils down to constructing an ϵ-net of small size with respect to $1/\epsilon$. Our strategy will be thus to compute such a small ϵ-net and then use either the Brönnimann and Goodrich algorithm [109] or the Even et al algorithm [110] as a blackbox. This will be our strategy in Section 3.3. Moreover, in Section 3.4, we will provide more advanced ϵ-net constructions for our purposes.

Depending on the underlying geometric objects in the set system, several ef-

ficient ϵ-net constructions have been developed. For example, we can look at the *Vapnik - Chervonenkis (VC)* dimension of the set system. The VC dimension of a set system is the size of the largest subset $Y \subseteq X$ that can be *shattered* in the sense that $|\{Y \cap S | S \in \mathcal{R}\}| = 2^{|Y|}$. When the set system has finite VC dimension d, Blumer et al [112] and Komlós et al [113] show that a random sample (under the probability distribution that assigns to $s \in X$ a probability $w(s)/w(X)$ of being sampled) of size $\mathcal{O}((\frac{d}{\epsilon}\log(\frac{1}{\epsilon})))$ turns out to be an ϵ-net with high probability. Better constructions are known for specific set systems: ϵ-nets of size $\mathcal{O}(1/\epsilon)$ are known for the case when the underlying objects are halfspaces in $\mathbb{R}^2$ or $\mathbb{R}^2$, pseudo-disks, fat wedges, three-sided axis-parallel rectangles in $\mathbb{R}^2$, and translates of quadrants in $\mathbb{R}^2$ and of fixed convex polytopes in $\mathbb{R}^3$ [1, 114–116].

When it comes to k-CENTER and k-SUPPLIERS, much less is known. Feder et al [117] have shown that it is NP-hard to approximate k-SUPPLIERS better than $\sqrt{7}$ and k-CENTER better than $\sqrt{3}$. This rules out the possibility of computing a PTAS for this case, making k-CENTER and k-SUPPLIERS one of the few clustering problems that do not admit arbitrarily close approximations in the Euclidean case. Nevertheless, this does not exclude the possibility of a $(2-\epsilon)$-approximation, for some $\epsilon > 0$. This remains one of the most interesting open problems in k-clustering. Recently, Nagarajan et al [118] have made great progress in this direction, by showing a $(1+\sqrt{3})$-approximation for Euclidean k-SUPPLIERS. It is unclear how this translates into an improved approximation for k-CENTER. In Section 3.5, we extend their result to work for the fault tolerant case.

Chapter 3

Clustering with Angular Constraints

In this chapter, we discuss the problem of placing a small number of sensors so as to satisfy the angular constraints defined in Section 1.2.1. We begin by reviewing existing work and summarizing our results and associated challenges (Section 3.1). We then formally define our problems and the necessary terms (Section 3.2). The main algorithmic framework can be found in Section 3.3. We then provide further approximations in Section 3.4 for the case of additional distance constraints. In Section 3.6, we discuss the computational complexity of the problem and provide evidence of hardness. Finally, we end with an approximation algorithm and some remarks for the case in which the input space is continuous (Section 3.7).

3.1 Introduction

In this section, we review previous works that directly relate to our problem, summarize our results in detail and discuss the challenges that arise along the way.

3.1.1 Related Work

When it comes to sensor coverage problems, extensive work has been done although surprisingly few results discuss α-coverage. Notable exceptions are the work of Efrat, Har-Peled and Mitchell [119], Tekdas and Isler [120] and Isler, Khanna, Spletzer and Taylor [121]. As mentioned before, Efrat et al. [119] introduce the α-AngArt problem in which two sensors are required to α-guard a target. They present a $\mathcal{O}(\log k_{\mathrm{OPT}})$-approximation algorithm that guarantees $\alpha/2$-coverage, where k_{OPT} is the smallest set of sensors that satisfy the visibility and α-coverage constraints. Their algorithm runs in time $\mathcal{O}(nk_{\mathrm{OPT}}^4 \log^2 n \log m)$. In contrast, we present a framework that achieves $(1 - 1/\delta) \cdot \alpha$-coverage.

Tekdas and Isler [120] formalize the angle constraint in a slightly different manner by considering an uncertainty function that depends not only on θ but also on the relative distance between sensors and the target: given two sensors s_1, s_2 and a target location t, the uncertainty is proportional to $\frac{d(s_1,t)\cdot d(s_2,t)}{|\sin\theta|}$, where d is the ℓ_2 metric. Notice that this corresponds exactly to the observation of Kelly [66]. They then investigate the problem of placing a minimum number of sensors such that the maximum uncertainty is below a given threshold U. A similar approach can

be used for the problem with just angular coverage (α-Ang) and for the one that additionally has distance constraints ((α, R)-AngDist).

When the targets are contained in some subset of the plane and the sensors can be placed anywhere, they present a 3-approximation with maximum uncertainty $\leq 5.5U$ for the case in which the uncertainty function is exactly $\frac{d(s_1,x)\cdot d(s_2,x)}{|\sin\theta|}$. This formulation applies directly to the task of target localization through triangulation and the method of Tekdas et al [120] captures this elegantly by imposing a triangular grid on the plane and placing sensors at vertices of the grid. By imposing a square grid however, we can show that the guarantee on the uncertainty can be met exactly, at the cost of employing a larger number of sensors. A similar approach can be used for α-Ang and (α, R)-AngDist.

Finally, Isler et al [121] consider the case in which the sensor locations are already given and one must compute an assignment of sensors to targets that minimizes the total sum of errors. In addition, they require that each sensor be used in tracking only one target. The version relevant to our problem is when the error is defined as $1/\sin\theta$. In the case in which the sensors are equally spaced on a circle, they present a 1.42-approximation that also applies to minimizing the maximum error.

3.1.2 Technical Challenges and Contributions

We introduce several new problems that require α-coverage and propose a general framework for approximating the angular constraint (Section 3.3). We ex-

emplify the use of this framework in the context of three specific problems: one in which we just consider angular constraints (α-Ang), one in which we have additional distance constraints ((α, R)-AngDist) and one in which we have visibility constraints (α-AngArt, also considered by Efrat et al [119]).

For the case of $\alpha \leq \pi/3$, we provide a general bi-criteria algorithm that approximates the angular coverage to arbitrary precision while guaranteeing a good approximation in the size of the solution. Specifically, for any $\delta > 1$, we propose an iterative method that guarantees $(1 - 1/\delta) \cdot \alpha$-coverage and contributes a factor of $\log \delta$ to the approximation factor of the solution size. A summary of our results can be found in Table 3.1.

In Section 3.4, we present further approximations for the special case in which we have angular and distance constraints. We relax the distance constraints from R to $2R$ and $3R$ and reduce the approximation factor of the solution size from $\mathcal{O}(\log \delta \cdot \log k_{\text{OPT}})$ to $\mathcal{O}(\log \delta)$ and $\mathcal{O}(1)$ respectively, while keeping the angular coverage at $(1 - 1/\delta) \cdot \alpha$. This emphasizes an inherent tradeoff in the problem formulation: if we allow our targets to be covered from farther away, we can achieve tighter guarantees on the number of sensors we pick.

We also consider the case in which $\alpha = 0$ and construct a set of optimal size that covers the targets within distance $(1 + \sqrt{3}) \cdot R$ (Section 3.5). This particular case remains relevant since it captures the spirit of fault tolerance by requiring two sensors to be assigned to a target. We achieve our result by showing a $(1 +$

Coverage Problem	Results
α-Ang	$\mathcal{O}(\log \delta)$-approximation with $(1 - 1/\delta) \cdot \alpha$-coverage
(α, R)-AngDist	within distance R: $\mathcal{O}(\log \delta \cdot \log k_{\text{OPT}})$-approx with $(1 - 1/\delta) \cdot \alpha$-coverage
	within distance $2R$: $\mathcal{O}(\log \delta \cdot \log \log k_{\text{OPT}})$-approx with $(1 - 1/\delta) \cdot \alpha$-coverage
	within distance $3R$: $\mathcal{O}(\log \delta)$-approx with $(1 - 1/\delta) \cdot \alpha$-coverage
	for $\alpha = 0$: optimal set of sensors within dist $(1 + \sqrt{3}) \cdot R$ Euclidean Fault Tolerant k-suppliers: Known 3-approximation [19], new $(1 + \sqrt{3})$-approx
α-AngArt	Known: $\mathcal{O}(\log k_{\text{OPT}})$-approx with $\alpha/2$-coverage [119]
	New: $\mathcal{O}(\log \delta \cdot \log k_{\text{OPT}})$-approx with $(1 - 1/\delta) \cdot \alpha$-coverage $\mathcal{O}(\log \delta \cdot \log h \cdot \log(k_{\text{OPT}} \log h))$-approx for h holes

Table 3.1: Summary of our results. Depending on each problem formulation, k_{OPT} denotes the size of the optimal set of sensors. The results hold for any $\delta > 1$ and $\alpha \leq \pi/3$.

$\sqrt{3}$)-approximation for the more general Euclidean Fault Tolerant k-suppliers problem which improves on the existing 3-approximation by Khuller et al [19].

In Section 3.7, we consider the original problem introduced by Tekdas et al [120] in which the sensors can be placed anywhere on the plane, the uncertainty function is given as $\frac{d(s_1,x) \cdot d(s_2,x)}{|\sin \theta|}$ and we are given an upper bound U on the maximum uncertainty. For this case, we give a 25-approximation that guarantees that the uncertainty is $\leq U$.

Lastly, we investigate the complexity of the angular constraint problem in Section 3.6. We begin by casting our problem as a Set Cover instance and show that the natural greedy strategy does not generally lead to good approximation algo-

rithms. We then abstract away the geometry of the angular constraints and define the Pairwise Selection problem. We show that this becomes a generalization of MinRep and is therefore hard to approximate. Finally, we show that the problem remains hard on planar graphs, from a reduction from Dominating Set.

Before describing our results however, we would like to outline the particular theoretical difficulties that the angular constraint gives rise to and provide some context for our strategy. Specifically, we will try to argue that classical techniques for Set Cover and k-center are not sensitive enough to accommodate for the particularities of the angular constraints and lead to poor approximations in general. We will focus on identifying what exactly makes this new model hard and then use this insight to develop a new technique that achieves improved approximations.

For simplicity, we will consider the α-Ang problem, noting that all our observations also apply to (α, R)-AngDist and α-AngArt, since they share the angular coverage constraint. One natural way in which we can consider α-Ang is as an instance of Set Cover. As will be seen later in the thesis (Section 3.6), this yields yields a $\mathcal{O}(k_{\text{OPT}} \cdot \log n)$- approximation for α-Ang. By exploiting the underlying geometry of the problem, we can improve the above approximation factor to $\mathcal{O}(k_{\text{OPT}} \log k_{\text{OPT}})$.

The persistent k_{OPT} factor in the approximation comes from the fact that the Set Cover framework cannot distinguish between sensors that help cover a lot of targets (in isolation) and sensors that, additionally, can also help cover more targets in conjunction with other sensors. In other words, it does not make use of the global dependency between sensors in order to get a small solution size. Such observations

are more in the vein of LABEL COVER type problems. In fact, as discussed in Section 3.6, when considered in its full generality (i.e. points lie in arbitrary space and coverage is defined arbitrarily), the problem becomes a generalization of MINREP and, as such, incurs a hardness of approximation bound of $2^{\log^{1-\epsilon} n}$, for any $0 < \epsilon < 1$ unless NP $\subseteq$ DTIME($n^{\text{polylog}(n)}$) [122]. This is, in fact, the **first bottleneck** of the problem: the contribution of one sensor depends on all the other sensors chosen in our solution. Better approximations must then exploit the fact that the contribution of one sensor is measured in terms of all the choices we make in our final solution and must therefore leverage the potential of already chosen sensors when selecting new ones.

A slightly different problem is encountered when considering classical methods for k-CENTER, especially in the fault tolerant setting of Khuller et al [19]. This observation specifically applies to the case in which we require each sensor to be within a range R of the target (as in (α, R)-ANGDIST). In Chapter 2, we discussed some techniques for k-center which boil down to arguing that far apart targets must be covered by distinct sensors. In the fault tolerant case, when the sensors collaborate to cover the targets, a similar approach can be used successfully. Furtermore, in the case of (α, R)-ANGDIST, the observations from before also hold.

The problem, however, comes from the fact that we can no longer make the argument that a target can be covered by any sensor at the price of paying for a larger distance. The choice of sensors that we make for those far apart targets might be optimal for those specific targets, but we cannot guarantee that they will help α-cover any of the other targets. When the sensors can be placed anywhere,

we can get around this problem by adding only a small constant number of new sensors (per each far apart target) that we can guarantee will cover the rest of the targets, wherever they might be. In contrast, when the sensors are restricted to only certain locations, we can no longer reason about which of the targets they will be able to cover. In general, each of the other targets might require two additional new sensors to be chosen in our solution. This, in turn, makes it hard to lower bound the size of the optimal solution and provide guarantees on the number of sensors the algorithm would choose. We identify this as the **second bottleneck** of our new model. Another way of thinking about this issue is that, in (α, R)-AngDist, the relative position of the sensors with respect to the target they are supposed to cover is encoded as a constraint rather than a parameter to be optimized (such as distance is in k-center). In this context, an approximation algorithm that relaxes that constraint would have to exploit its inherent geometric structure so that it can cover the rest of the targets in a way that can be translated quantitatively as a function of the size of the optimal solution.

In this context, the main contribution in this chapter is a general technique for constructing solutions with approximate α-coverage when $\alpha \leq \pi/3$. In particular, we present a method that, given a set of sensors S that achieves $(\alpha-2\epsilon)$-coverage, builds a slightly larger set S' that obtains $(\alpha - \epsilon)$-coverage, for any $\epsilon \leq \alpha/2$. Essentially, $(\alpha - \epsilon)$-coverage will be obtained by constructing pairs in which one sensor is in S and the other one is in S', thus reducing the dependency between sensors and limiting the set of pairs we consider in each step. Specifically, using the fact that S already achieves $(\alpha - 2\epsilon)$-coverage, we reduce the task of constructing S' to that of

computing hitting sets for set systems induced by geometric objects for which good approximation factors exist. At this point, any target that is not yet $(\alpha - \epsilon)$-covered by the sensors in S will be $(\alpha - \epsilon)$-covered by one sensor in S and another sensor in S'. Moreover, we can bound the size of S' in terms of the optimal solution size and therefore consistently keep track of our performance guarantee.

The method generalizes the one used by Efrat et al [119] and can be iteratively applied to obtain better and better angular guarantees at the expense of constructing a larger set each time. It is worthwhile to note that the main technical contribution of the proposal refers to the angle coverage constraint and as such, could be applied to a variety of other problems as long as the other constraints (such as distance or line-of-sight visibility) define a good set system (one with finite VC dimension, for example).

For a given $\delta > 0$, our method ensure an angular coverage of $(1 - \delta) \cdot \alpha$ and an approximation factor for the number of sensors used that depends linearly on the number of iterations, only logarithmically on k_{OPT} and does not directly depend on $\log \delta$. Specifically, when the framework is used a constant number of times, we get a $\mathcal{O}(1)$-approximation for α-Ang and $\mathcal{O}(\log k_{\mathrm{OPT}})$-approximations for (α, R)-AngDist and α-AngArt, while approximating the angle coverage by a constant. This improves on the result by Efrat et al. [119] for α-AngArt for the case of $\alpha \leq \pi/3$.

3.2 Problem Formulation and Definitions

Let $X \subseteq \mathbb{R}^2$ be the set of potential sensor locations and $T \subseteq \mathbb{R}^2$ be the set of target locations. The underlying distance function will be the ℓ_2 metric.

We consider (unordered) pairs of the form (s, s') where $s \neq s'$, $s \in S$, $s' \in S'$ and $S, S' \subseteq X$. We denote the set of such pairs as $S \times S'$. Formally,

$$S \times S' = \{(s, s') | s \neq s', s \in S, s' \in S'\}.$$

Definition 3.1. *We say that the pair (s, s') α-covers the target t if the angle $\angle sts' \in [\alpha, \pi - \alpha]$.*

We also say that the set of pairs $S \times S'$ α-covers t if there exists a pair $(s, s') \in S \times S', s \neq s'$, that α-covers t. When $S' = S$, we simply say that the set S α-covers t. We say that a pair or a set of pairs α-covers a set T of targets when it α-covers each element of T. Notice that the parameter α defines a certain range: the higher the value of α, the smaller the range of possible values that $\angle sts'$ can take. We are now ready to define α-Ang:

Min SPP with Angle Constraints (α-Ang)

Input : A set of sensor locations X, a set T of target locations, an angle parameter $\alpha \in [0, \pi/2]$.

Output : A set $S \subseteq X$ of mininum cardinality such that S α-covers T.

In addition, we can require that the sensors be within a range of the targets they cover:

Definition 3.2. *We say that a pair or a set of pairs α-covers a set T of targets within distance R if, for at least one of the pairs that α-covers a target, the distance from both sensors to the target is $\leq R$.*

We are now ready to define (α, R)-AngDist:

Min SPP with Angle and Distance Constraints ((α, R)-AngDist)
Input : A set of sensor locations X, a set T of target locations, an angle parameter $\alpha \in [0, \pi/2]$ and a distance $R \geq 0$.
Output : A set $S \subseteq X$ of minimum cardinality such that S α-covers T within distance R.

Finally, we consider the problem introduced by Efrat et al [119] which combines angle coverage with line-of-sight constraints. Given two regions $Q \subseteq P$, the goal is to place a small set of sensors in P that guard targets in Q. In order to obtain a discrete set of sensors, the authors impose an arbitrary grid Γ on P and consider potential sensors locations that are situated at the vertices of Γ. We note however, that such a step is not necessary in our case, since X is already a discrete set. We also require the set of target locations to be inside Q, $T \subseteq Q$. Given a sensor $s \in X$ and a target $t \in T$, we say that s sees t if the segment connecting the two does not cross the boundary of P.

Definition 3.3. *Two sensors s_1, s_2 α-guard a target t if they both see the target and (s_1, s_2) α-covers t.*

By extension, we say that a set $S \subseteq X$ α-guards T if, for each target $q \in T$, there exist sensors $s_1, s_2 \in X$ such that (s_1, s_2) α-guards q. We get the following

optimization problem:

Min SPP with Angle and Line-of-Sight Constraints (α-AngArt)
Input : Polygons $Q \subseteq P$,$X \subseteq P$, $T \subseteq Q$, and an angle parameter $\alpha \in [0, \pi/2]$.
Output : A set $S \subseteq X$ of minimum cardinality such that S α-guards T.

From now on, we will refer to an optimal solution to each of the proposed problems as S_{OPT} and to its cardinality as k_{OPT}. The exact problem these quantities refer to will be obvious from context. We will also consider α and R to be global parameters.

3.3 Algorithmic Framework for α-coverage

In this section, we describe an algorithm that, given a set S that $(\alpha - 2\epsilon)$-covers T, constructs a set S' that $(\alpha - \epsilon)$-covers T. We refer to this part as the *angle reduction* step because it produces a solution with a smaller angle relaxation. First, we will show that the set S' exists and that its size is upper bounded by k_{OPT} (Section 3.3.1). Specifically, we will show that $S \times S_{\text{OPT}}$ $(\alpha - \epsilon)$-covers T. The first observation in this line of thought was made by Efrat et al [119]: given an arbitrary set $S \subseteq X$, for every target $t \in T$, there exists a sensor $s \in S$ and $s^* \in S_{\text{OPT}}$ such that (s, s^*) $\alpha/2$-covers t. In other words, $S \times S_{\text{OPT}}$ $\alpha/2$-covers T. We generalize this observation for the case when $\alpha \leq \pi/3$:

Lemma 3.1. *Let $\epsilon > 0$ be such that $\alpha - \epsilon \leq \pi/3$ and $\epsilon \leq \alpha/2$. Given a set S that $(\alpha - 2\epsilon)$- covers T, let $T' \subseteq T$ be the set of targets that S does **not** $(\alpha - \epsilon)$-cover. Then $S \times S_{\text{OPT}}$ $(\alpha - \epsilon)$-covers T' (i.e. the set $S \cup S_{\text{OPT}}$ $(\alpha - \epsilon)$-covers T).*

In other words, if S does not already $(\alpha - \epsilon)$-cover a target, then the sensors in S can be paired up with sensors in S_{OPT} to $(\alpha - \epsilon)$-cover it. When $\epsilon = \alpha/2$, we start with an arbitrary set S and recover the observation of Efrat et al. [119] for $\alpha \leq \pi/3$. We note that, in order to get a better that 1/2-approximation on the angular coverage, the seed set S cannot be chosen arbitrarily. In fact, our proof crucially uses the power of the pairs in S to $(\alpha - 2\epsilon)$-cover the targets, thus further exploiting the collaboration between already chosen sensors.

We also note that S_{OPT} is in fact more powerful that this since it α-covers everything. The problem, however, is that we do not know how to fully exploit the power of the pairs in $S_{\text{OPT}} \times S_{\text{OPT}}$. By fixing some of the sensors to be in S and looking at pairs in $S \times S_{\text{OPT}}$, however, we can reduce the general problem to a more tractable one of finding a suitable set S' that can achieve what S_{OPT} achieves in this restricted framework.

Finally, notice that the above lemma only provides an existential proof for S', since we do not know what S_{OPT} is. We then show how to construct S' for each of the problems considered (Section 3.3.2). In particular, we exploit this restricted framework and construct the set S' by computing an approximate hitting set. We then show how to use the algorithm iteratively and obtain $(1 - 1/\delta) \cdot \alpha$-coverage (Section 3.3.3).

3.3.1 Existence of an Approximate Solution

We begin by defining, for each target $t \in T$, sensor $s \in X$, angle parameter $\beta \in [0, \pi/2]$ and distance $r \geq 0$, the set

$$R_t(s, \beta) = \{s' \in X | (s, s') \ \beta\text{-covers } t\}.$$

In other words, $R_t(s, \beta)$ represents the set of feasible locations for sensors that, together with s, β-cover t (β will be instantiated as $\alpha - \epsilon$). Notice that, from the point of view of t, the set $R_t(s, \beta)$ is induced by two wedges around t, as seen in Figure 3.1. A *wedge* is defined as the intersection of two non-parallel halfspaces in $\mathbb{R}^2$. Specifically, let l be the line that passes through s and t. Let l_1 and l_2 be the two lines that pass through t and form an angle of β with l. These two lines describe two opposite wedges of interest: one that corresponds to the halfspaces above the lines l_1 and l_2, and one that corresponds to the halfspaces below the lines l_1 and l_2. These two wedges are exactly the ones that describe the feasible region where s' could be and we shall refer to their union as a *double-wedge* from now on. The central angle θ of both the wedges is $\theta = \pi - 2\beta$, and by extension we shall say that the corresponding double-wedge has a central angle of $\theta = \pi - 2\beta$.

In order to prove Lemma 3.1, we essentially show that S_{OPT} must intersect the double-wedges generated by S and T. First, notice that if a set S already $(\alpha - \epsilon)$-covers a target t, then we do not need to worry: S will continue to $(\alpha - \epsilon)-$cover t even when we add S' to S. We are therefore concerned with targets in T' that are not already $(\alpha - \epsilon)$-covered by S. We note, however, that it is essential that S $(\alpha - 2\epsilon)$-covers T'. If S does not have this property or is arbitrary, we cannot hope

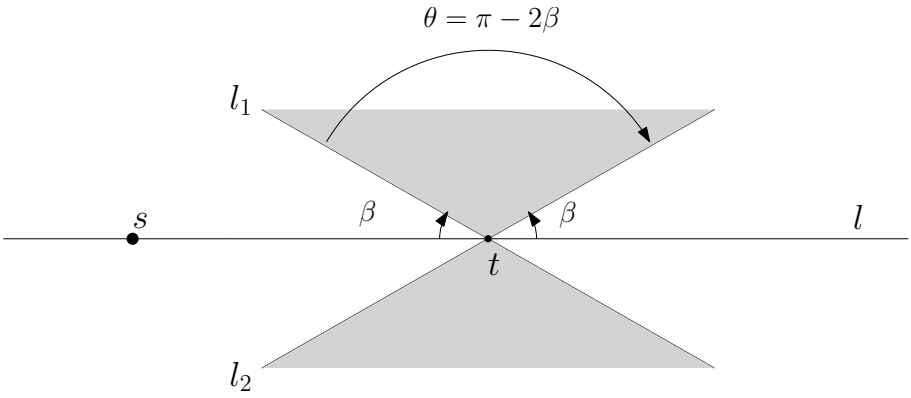


Figure 3.1: The set $R_t(s, \beta)$ is induced by the double-wedge generated by the lines l_1 and l_2 and has a central angle $\theta = \pi - 2\beta$.

to get past the $\alpha/2$ barrier.

Fix such a target $t \in T'$ and let $s_1, s_2 \in S$ be any two sensors that $(\alpha - 2\epsilon)$-cover t but do not $(\alpha - \epsilon)$-cover it. We will show that there exists $s^* \in S_{\text{OPT}}$ such that either $s^* \in R_t(s_1, \alpha - \epsilon)$ or $s^* \in R_t(s_2, \alpha - \epsilon)$. The candidates will be $s_1^*, s_2^* \in S_{\text{OPT}}$ where (s_1^*, s_2^*) is the optimal pair that α-covers t. Intuitively, each of the double-wedges induced by s_1 and s_2 alone is not big enough to "capture" s_1^* or s_2^*. However, if $\angle s_1 t s_2$ is in the range $[\alpha - 2\epsilon, \pi - (\alpha - 2\epsilon)]$, then together, the union D_t of these double-wedges (which will also be a double-wedge around t) will be sufficiently well spread (i.e. have a large enough central angle) to guarantee that one of the optimal sensors is contained in it. In other words, at least one of the optimal sensors s_1^* or s_2^* together with either s_1 or s_2 will $(\alpha - \epsilon)$-cover t.

We note, however, that the requirement that α be smaller than $\pi/3$ is relatively tight in this framework, in the sense in which, if $\alpha - \epsilon > \pi/3$, then the central angle of each of the double wedges is too small and we can no longer guarantee that their union D_t forms a bigger double wedge. Furthermore, it is not true that D_t must

intersect S_{OPT}.

Consider the double-wedges D_1 and D_2 corresponding to $R_t(s_1, \alpha - \epsilon)$ and $R_t(s_2, \alpha - \epsilon)$, respectively, with central angles $\theta_{D_1} = \theta_{D_2} = \pi - 2(\alpha - \epsilon)$. Let $\alpha' = \angle(s_1 t s_2)$.

Lemma 3.2. *The union of the two double-wedges D_1 and D_2 is a larger double-wedge D centered at t with central angle $\theta_D = \pi - 2(\alpha - \epsilon) + \alpha'$.*

Proof. We refer the reader to Figure 3.2 for an intuitive explanation. Formally, let l be the line that passes through s_1 and t and let l_1 and l_2 the two lines that define D_1. Since $\angle(s_1 t s_2) \notin [\alpha - \epsilon, \pi - (\alpha - \epsilon)]$, it follows that s_2 is not in D_1. Assume without loss of generality that s_2 is between the lines l and l_1 in the counterclockwise direction. The same proof follows for the other possible locations of s_2.

Now consider D_2 and let l_3 and l_4 be the defining lines through t, while l' is the line that passes through s_2 and t. Notice that D_1 and D_2 are identical except that D_2 is a rotated copy of the D_1. In other words, since $\angle(l, l') = \alpha'$, we also have that $\angle(l_1, l_3) = \alpha'$ and $\angle(l_2, l_4) = \alpha'$.

Furthermore, since $\alpha' \leq \alpha - \epsilon$ and $\angle(l_1, l_3) \leq \pi - 2(\alpha - \epsilon)$, we have that when $\alpha - \epsilon < \pi/3$, l_3 lies in between l_1 and l_2 and the union of the two double-wedges D_1 and D_2 is a continuous double-wedge D determined by l_1 and l_4. It has central angle $\theta_D = \theta_1 + \angle(l_2, l_4) = \pi - 2(\alpha - \epsilon) + \alpha'$. □

Our goal is to show that one of the two optimal sensors s_1^* and s_2^* must be in D_t. The intuition is that by making D_t have a large central angle, we ensure that the complement D_t' of D_t has such a small central angle that it would not be able

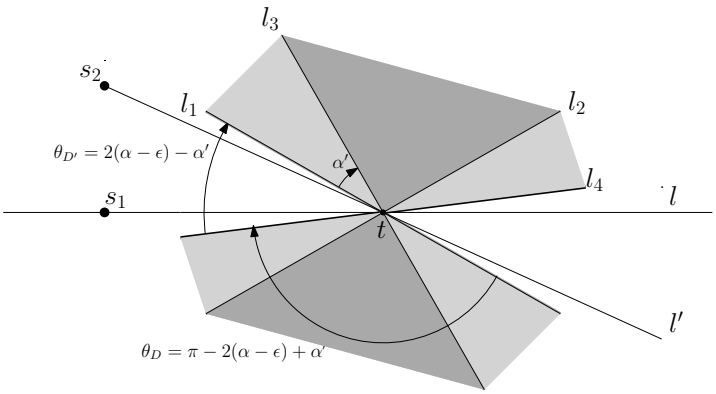


Figure 3.2: Since $\angle(s_1 t s_2) = \theta$, D_2 is a rotation by α' of D_1 . Their union is another double-wedge D defined by l_1 and l_4 with central angle $\theta_D = \pi - 2(\alpha - \epsilon) + \alpha'$.

to contain both s_1^* and s_2^*.

Lemma 3.3. *One of the two optimal sensors s_1^* and s_2^* must be in D.*

Proof. We refer the reader to Figure 3.3 for an intuitive explanation. Let D' be the complement of D. Notice that D' forms another double-wedge defined by l_1 and l_4 but that it does not actually contain points on these lines. The intuition is that by making D have a large central angle, we ensure that D' has such a small central angle that it would not be able to contain both s_1^* and s_2^*.

First, notice that D' has a central angle

$$\theta_{D'} = \pi - \theta_D = 2(\alpha - \epsilon) - \alpha'.$$

Since s_1 and s_2 $(\alpha - 2\epsilon)$-cover the target, and we are considering the case where s_2 is between l and l_1, we have:

$$\alpha' \geq \alpha - 2\epsilon.$$

Hence, we have that $\theta_{D'} \leq \alpha$. This implies that s_1^* and s_2^* cannot be both in the same wedge of D' without being exactly situated on the lines l_1 and l_4(i.e. in

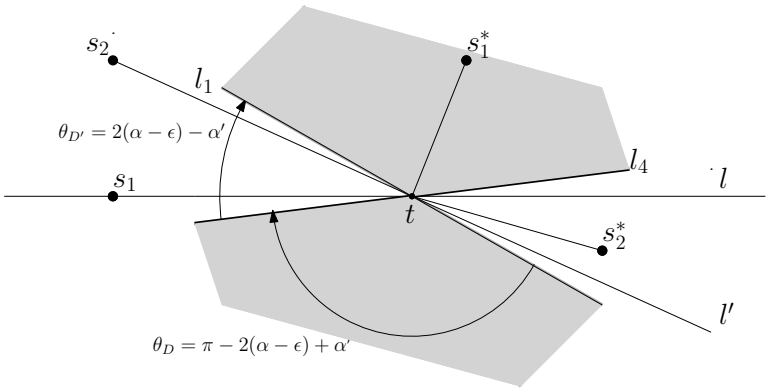


Figure 3.3: We want to show that one of the optimal sensors s_1^*, s_2^* must be in D. Remember that, by definition, $\angle s_1^* t s_2^* \in [\alpha, \pi - \alpha]$. Suppose by contradiction that they are not in D. Then they must be in D'. If they are both in the same wedge of D', then we must have that $\angle s_1^* t s_2^* \leq \theta_{D'}$. But $\theta_{D'} \leq \alpha$. On the other hand, if they are in different wedges of D', then $\angle s_1^* t s_2^* \geq \theta_D$. But $\theta_D \geq \pi - \alpha$.

D).

The other bad situation would be for them to be in different wedges of D'. But then the angle between them would be greater than θ_D. Since $\alpha' \geq \alpha - 2\epsilon$, we get that

$$\theta_D = \pi - 2(\alpha - \epsilon) + \alpha' \geq \pi - \alpha,$$

which would contradict the fact the $\angle(s_1^* t s_2^*) \in [\alpha, \pi - \alpha]$. In other words, at least one of the optimal sensors s_1^* and s_2^* must be in D. □

3.3.2 Construction of the Approximate Solution

Once we have determined that S_{OPT} satisfies the requirements of Lemma 3.1, we will show how to construct a set S' of approximate size that also $(\alpha - \epsilon)$-covers T'. As noted before, the proof of Lemma 3.1 only talks about angle coverage. Depending on the problem at hand, the construction of S' will differ but the general technique

is the same. We first illustrate it for α-Ang and then mention how to change it for (α, R)-AngDist and α-AngArt.

In the proof of Lemma 3.1, for each target $t \in T'$ and pair $(s_1, s_2) \in S \times S$ that $(\alpha - \epsilon)$-covers it, we defined a double-wedge that we showed must contain one of the sensors in S_{OPT}. Let D_t be that double-wedge, $\forall t \in T'$. Consider the set system $\mathcal{F}(X, \mathcal{R})$ induced on X by these double-wedges, where we define:

$$\mathcal{R} = \{D_t \cap X |\ t \in T'\}.$$

Notice that S_{OPT} intersects every D_t non trivially. In other words:

Lemma 3.4. *The set S_{OPT} is a hitting set for $\mathcal{F}(X, \mathcal{R})$.*

In general, a set of sensors $H \subseteq X$ is a *hitting set* for $\mathcal{F}(X, \mathcal{R})$ if

$$H \cap D_t \neq \emptyset,\ \ \forall\, t \in T'.$$

In other words, if we want to construct S', we can compute a hitting set for $\mathcal{F}(X, \mathcal{R})$. In this context, the Hitting Set problem asks for a hitting set of minimum cardinality. Let τ be the size of the optimal set. Notice that Lemma 3.1 shows that S_{OPT} is a hitting set for $\mathcal{F}(X, \mathcal{R})$, so we are guaranteed that $\tau \leq k_{\text{OPT}}$. Our strategy therefore will be to construct S' as an approximate solution to the Hitting set problem for $\mathcal{F}(X, \mathcal{R})$ and obtain a guarantee on S' in terms of k_{OPT}.

A particularly simple yet elegant ϵ-net construction was given by Kulkarni and Govindarajan [1] for the case of γ-fat wedges. A *γ-fat wedge* is a wedge having a central angle of at least γ. When the sets in $\mathcal{R}$ are induced by such γ-wedges, Kulkarni and Govindarajan [1] construct an ϵ-net of size $\mathcal{O}(\frac{\pi}{\gamma\epsilon})$ for arbitrary γ.

When $\gamma \geq \pi/2$, the size of the ϵ-net becomes $\mathcal{O}(\frac{1}{\epsilon})$. This directly applies to the α-Ang problem because the wedges in each double-wedge have a central angle $\theta_D = \pi - 2(\alpha - \epsilon) + \alpha' \geq \pi - \alpha$, since $\alpha' \geq \alpha - 2\epsilon$. Therefore, they are $(\pi - \alpha)$-fat. Each double-wedge can be decomposed into two disjoint wedges, so an $\frac{\epsilon}{2}$-net for $(\pi - \alpha)$-fat wedges is guaranteed to be an ϵ-net for our double-wedges. Since $\pi - \alpha \geq \pi/2$, we get a hitting set of size $\mathcal{O}(\tau)$. Adding this set to S, we get the following:

Lemma 3.5. *(**Angle Reduction Step for** α**-Ang**) Let $\epsilon > 0$ be such that $\alpha - \epsilon \leq \pi/3$ and $\epsilon \leq \alpha/2$. Let $S \subseteq X$ be a set that $(\alpha - 2\epsilon)$- covers T. Then we can find a set $S' \subseteq X$ such that S' $(\alpha - \epsilon)$-covers T and $|S'| = |S| + \mathcal{O}(1) \cdot k_{\mathrm{OPT}}$. The running time of the algorithm is $\mathcal{O}(k_{\mathrm{OPT}} \cdot m \log m)$, where $m = |X|$.*

When we consider the (α, R)-AngDist problem, the set of feasible sensor locations becomes the intersection of a double-wedge with the circle of radius R centered at the corresponding target. Notice that while the double-wedge captures the angle requirements (as it did for α-Ang), the circle represents the additional distance constraints (specific to (α, R)-AngDist). Hence, for each target t, we define the double-sector

$$C_t = D_t \cap \mathcal{C}(t, R).$$

The appropriate set system then becomes $\mathcal{F}'(X, \mathcal{R}')$, where

$$\mathcal{R}' = \{C_t \cap X |\ t \in T'\}.$$

Similar to double-wedges, each double-sector is composed of two disjoint sectors and hence, an $\frac{\epsilon}{2}$-net for sectors would be an ϵ-net for double-sectors. Since

each sector is the intersection of two halfspaces and a circle, each of which induce set system that have constant VC-dimension, we get that it also has constant VC-dimension [123]. Therefore, it admits an ϵ-net of size $\mathcal{O}((\frac{d}{\epsilon}\log(\frac{1}{\epsilon})))$ [112, 113]. That, in turns, gives us a hitting set of size $\mathcal{O}(d\tau \cdot \log(\tau))$ for $\mathcal{F}'(X, \mathcal{R}')$. As before, adding this set to S, we get the following:

Lemma 3.6. *(**Angle Reduction Step for** (α, R)**-AngDist**) Let $\epsilon > 0$ be such that $\alpha - \epsilon \leq \pi/3$ and $\epsilon \leq \alpha/2$. Let $S \subseteq X$ be a set that $(\alpha - 2\epsilon)$- covers T within distance $R' \geq 0$. Then we can find a set $S' \subseteq X$ such that S' $(\alpha - \epsilon)$-covers T within distance $\max\{R, R'\}$ and $|S'| = |S| + \mathcal{O}(\log k_{\mathrm{OPT}}) \cdot k_{\mathrm{OPT}}$. The running time of the algorithm is $\mathcal{O}(k_{\mathrm{OPT}} \cdot mn \log m)$, where $n = |T|$ and $m = |X|$.*

Notice that in the above statement, the distance within which S covers the target set does not have to be R. This does not affect our hitting set construction. This later observation will prove useful when we relax R' to be $2R$ in Section 3.4.2.

In the case of α-AngArt, for each target t, the appropriate set system $\mathcal{F}''(X, \mathcal{R}'')$ is built by intersecting D_t with the set of sensors V_t in X that guard t, called the visibility polygon of t. When the underlying polygon P is simply connected, the set system $(X, \{V_t \cap X |\ t \in T'\})$ has constant VC-dimension, as pointed out by Valtr [124]. It follows that $\mathcal{F}''(X, \mathcal{R}'')$ also has finite VC-dimension [123]. As such, it has a hitting set of size $\mathcal{O}(k_{\mathrm{OPT}} \log(k_{\mathrm{OPT}}))$. As before, adding this set to S, we get the following:

Lemma 3.7. *(**Angle Reduction Step for** α**-AngArt**) Let $\epsilon > 0$ be such that $\alpha - \epsilon \leq \pi/3$ and $\epsilon \leq \alpha/2$. Let $S \subseteq X \subseteq P$ be a set that $(\alpha - 2\epsilon)$- guards $T \subseteq Q \subseteq P$.*

When P is a simply connected polygon, we can find a set $S' \subseteq X$ such that S' $(\alpha-\epsilon)$-covers Q and $|S'| = |S| + \mathcal{O}(\log k_{\mathrm{OPT}}) \cdot k_{\mathrm{OPT}}$. The running time of the algorithm is $\mathcal{O}(k_{\mathrm{OPT}} \cdot mn \log m)$, where $n = |T|$ and $m = |X|$.

3.3.3 Iterative Algorithm

Given the technical lemmas from before that allow us to refine the angular coverage of a given seed set S, we can now develop a more general algorithm that constructs a new set that achieves $((1 - 1/\delta) \cdot \alpha)$-coverage for any $\delta > 1$.

The idea is to iteratively apply the angle reduction step $\log \delta$ times, first with $\epsilon_1 = \alpha/2$, then with $\epsilon_2 = \alpha/4$ etc. Let S_i be the set at the beginning of iteration $i \in \{1, \ldots, \log \delta\}$. S_i will correspond to the set S in the angle reduction step. Once we employ the appropriate construction, we obtain a set S' and set $S_{i+1} = S' \cup S$. The invariant we will maintain is that, in iteration $i > 1$, the set S_i $((1 - 1/2^i)) \cdot \alpha$-covers T. This can be easily shown by induction on i. At the end of $\log \delta$ iterations, we have that the set $S_{\log \delta + 1}$ $((1 - 1/\delta) \cdot \alpha)$-covers T.

The running time of the algorithm is $\log \delta$ times the time to find the appropriate hitting set plus the time it takes to find S_1. This first set requires special care and depends on the problem at hand. We require S_1 to 0-cover T but one can check that the proof of Lemma 3.1 follows in this case even when we do not have two distinct sensors covering a target. Therefore, in the case of α-Ang, it suffices to pick S_1 to consist of any sensor in X. In each iteration, we increase the size of our set by $\mathcal{O}(1) \cdot k_{\mathrm{OPT}}$ and since $|S_1|$=1, we get the following:

Theorem 3.1. *Given X, T, $\alpha \in [0, \pi/3]$ as above, we can find a set of sensors $S \subseteq X$ such that S $((1 - 1/\delta) \cdot \alpha)$-covers T and $|S| = \mathcal{O}(\log \delta) \cdot k_{\text{OPT}}$, where k_{OPT} is the cardinality of the smallest set of sensors that α-covers T. The running time of the algorithm is $\mathcal{O}(k_{\text{OPT}} \cdot m \log \delta \log m)$, where $n = |T|$ and $m = |X|$.*

When it comes to the (α, R)-AngDist problem, we require that the initial set S_1 has the property that each target is within distance R of at least one sensor in S_1. Notice that this is the analogue of the r-domination problem for the case of (k, r)-suppliers (i.e. when the centers must be picked from a distinct set). Moreover, without loss of generality, we can assume that $R = 1$ and then our problem becomes an instance of the Discrete Unit Disk Cover (DUDC) problem. In DUDC, we are given a set $\mathcal{P}$ of n points and a set of $\mathcal{D}$ of m unit disks in the Euclidean plane. The objective is to select a set of disks $\mathcal{D}^* \subseteq \mathcal{D}$ of minimum cardinality that covers all the points. The problem is a geometric version of Set Cover and is NP-hard [92]. Nevertheless, several constant factor approximations have been developed. For our purpose, we will use the 18-approximation by Das et al [125] that has a runtime of $\mathcal{O}(n \log n + m \log m + mn)$. We note that better approximations are known, but using them in our framework would increase the total runtime considerably. In each iteration, we increase the size of our set by $\mathcal{O}(\log k_{\text{OPT}}) \cdot k_{\text{OPT}}$ and since $|S_1| \leq 18 \cdot k_{\text{OPT}}$, we get the following:

Theorem 3.2. *Given X, T, α and R as above, we can find a set of sensors $S \subseteq X$ such that S $((1 - 1/\delta) \cdot \alpha)$-covers T within distance R and $|S| = \mathcal{O}(\log n \log k_{\text{OPT}}) \cdot k_{\text{OPT}}$, where k_{OPT} is the cardinality of the smallest set of sensors that α-covers T*

within distance R. The running time of the algorithm is $\mathcal{O}(k_{\mathrm{OPT}} \cdot mn \log m \log \delta)$, where $n = |T|$ and $m = |X|$.

For α-AngArt, we need to find a set of sensors $S_1 \subseteq X$ that guard T. To this extent, we again employ the fact that the set of visibility polygons has finite VC-dimension [124]. Notice that finding a small S_1 that guards T is the hitting set problem for the set system made of sensors and visibility polygons corresponding to each target. We therefore obtain a set of size $\mathcal{O}(\log k^*) \cdot k^*$, where k^* is the size of the smallest set of sensors from X that guard T. Notice that, since S_{OPT} also guards T, we have that $k^* \leq k_{\mathrm{OPT}}$. The running time of the algorithm is $\mathcal{O}(nk_{\mathrm{OPT}}^2 \cdot \log n \log(nk_{\mathrm{OPT}}))$. In each iteration, we increase the size of our set by $\mathcal{O}(\log k_{\mathrm{OPT}}) \cdot k_{\mathrm{OPT}}$ and since $|S_1| = \mathcal{O}(\log k_{\mathrm{OPT}}) \cdot k_{\mathrm{OPT}}$, we get the following:

Theorem 3.3. *Given polygons $Q \subseteq P$,X, T, and $\alpha \in [0, \pi/3]$ as above, we can find a set of sensors $S \subseteq X$ such that S $((1 - 1/\delta) \cdot \alpha)$-guards T and $|S| = \mathcal{O}(\log n \log k_{\mathrm{OPT}}) \cdot k_{\mathrm{OPT}}$, where k_{OPT} is the cardinality of the smallest set of sensors that α-guards T. The running time of the algorithm is $\mathcal{O}(k_{\mathrm{OPT}} \cdot mn \log m \log n)$, where $n = |T|$ and $m = |X|$.*

Notice that we can also apply the algorithm only a constant number of times. In particular, for any $c > 0$, we get the following results:

- for α-Ang, we get a $\mathcal{O}(1)$-approximation that achieves $((1-1/2^c)\cdot\alpha)$-coverage and runs in time $\mathcal{O}(k_{\mathrm{OPT}} \cdot m \log m)$
- for (α, R)-AngDist and α-AngArt, we get a $\mathcal{O}(\log k_{\mathrm{OPT}})$-approximation that runs in time $\mathcal{O}(k_{\mathrm{OPT}} \cdot mn \log m)$.

We note that, in the case of α-AngArt, we obtain better approximations that Efrat et al [119], in the sense that we are able to approximate the α-coverage constraint to any constant factor while using only $\mathcal{O}(\log k_{\text{OPT}} \cdot k_{\text{OPT}})$ sensors. Moreover, our running times are comparable: $\mathcal{O}(nk_{\text{OPT}}^4 \log^2 n \log m)$ in Efrat et al versus $\mathcal{O}(k_{\text{OPT}} \cdot mn \log m)$ for our approximation. We note that their running time comes from the fact they do not directly use the bounded VC dimension of the set system. Instead, they use a previous algorithm designed by Efrat et al [126] for approximating the Art Gallery problem when the set of targets is unconstrained. In that scenario, they impose a grid on the space of targets and restrict their choices to vertices of the grid. The intuition is that this is a good approximation for the general case when we can place targets anywhere and a good assumption since it allows them to design a Brönnimann and Goodrich algorithm [109] - type algorithm. For this restricted problem, they obtain a $\mathcal{O}(\log k^*)$-approximation, where k^* is the size of the optimal set of sensors places on vertices of the grid that could guard the entire polygon. The expected algorithm runs in time $\mathcal{O}(nk^{*2} \log k^* \log(nk^*) \log^2 \Delta)$, where Δ is the ratio between the diameter of the polygon and the grid size. When angle constraints are added, they adapt this algorithm to only consider vertices of the grid that also satisfy α-coverage.

In our scenario in which targets have to be chosen from a discrete set, we do not need to impose a grid and can directly apply the Brönnimann and Goodrich algorithm [109] algorithm. For the case in which the targets can be placed anywhere, their algorithm could be employed instead while maintaining the same approximation guarantees.

3.4 Relaxed Distance Constraints and α-coverage

The geometric objects at the core of our method are wedges centered at targets whose central angles depend on α and ϵ. These ranges define the set of feasible locations from which we must choose a new set of sensors and are given as input to the afferent hitting set problem. In the case of (α, R)-AngDist, the distance constraints require that the sensors we pick be within range R of the target, so our wedges become sectors through intersection with a disk of radius R centered at the target. In this context, we employ the canonical ϵ-net construction of Blumer et al [112] and Komlós et al [113] and obtain a $\mathcal{O}(\log k_{\mathrm{OPT}})$-approximation.

In an attempt to reduce the approximation factor in this latter case, we consider relaxing the distance constraint and allowing the chosen sensors to be within distance $2R$ or $3R$ of the targets. Relaxing the distance allows us to include these double-sectors in unions of objects for which better ϵ-net constructions exist. The question then becomes which secondary objects we should consider and how to design specific ϵ-net for them.

To this end, we will first review known constructions and identify challenges we might encounter when trying to adapt them for our double-sectors. In the following sections, we will discuss our constructions in detail and show that they lead to good bicriteria approximations. We note an interesting phenomena, however:

- without violating the distance contraints, we can get a $\mathcal{O}(\log k_{\mathrm{OPT}})$-approximation on the size of the hitting set,

- with violating the constraint to $2R$, we can get a $\mathcal{O}(\log \log k_{\mathrm{OPT}})$-approximation,
- with violating the constraint to $3R$, we can get a $\mathcal{O}(1)$-approximation.

Our framework therefore explores the inherent trade-off in the optimization component: once we allow our targets to be covered from farther away, we can pick less sensors to cover them. Moreover, we believe the methods used to achieve them might be of independent interest for other problems in which the sets to be hit by the ϵ-net are induced by targets that need to be covered within a specific distance.

3.4.1 Challenges

In general, one possible technique for building a good ϵ-net is to construct a grid-like structure on top of the input points and pick points (deterministically or randomly) that will be guaranteed to intersect the heavy ranges. Without any guarantees on the sizes of the ranges (for some notion of size), one could split the input points into vertical and horizontal strips that each contain some constant fraction of $\epsilon \cdot n$ points and pick points in each such cell. The heavy ranges will be then guaranteed to intersect at least a constant fraction of these cells, which could then be used to show that they also contain at least one of the chosen points. The direct consequence of allowing both such a horizontal and vertical discretization, however, would be that the number of cells would increase quadratically in $\frac{1}{\epsilon}$, leading to sub optimal bounds on the size of the ϵ-net. Better constructions would have to somehow allow such a discretization to happen in only one direction and present efficient techniques for picking less that $1/\epsilon$ points in each such cell. As an example,

we refer the reader to the result of Aronov et al. [127] for a powerful construction that builds an ϵ-net of size $\mathcal{O}(\frac{1}{\epsilon}\log\log(\frac{1}{\epsilon}))$ for a variety of objects such as axis-parallel rectangles, boxes and α-fat triangles.

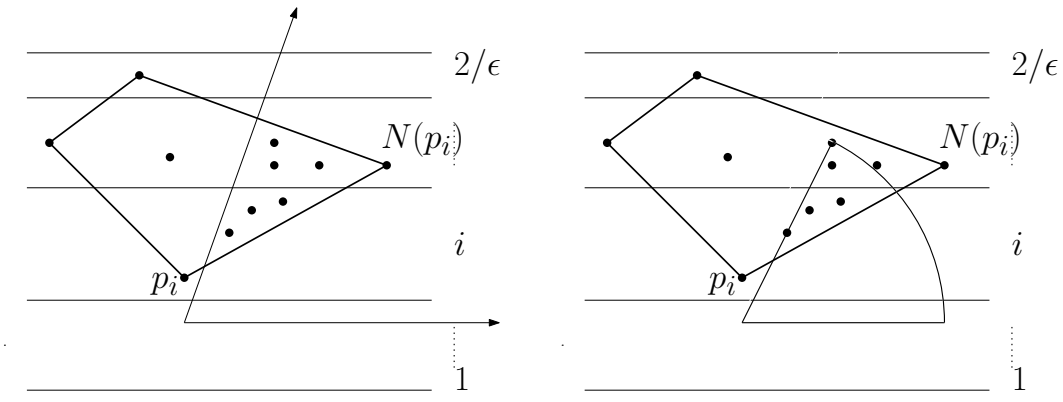
In this context, the elegant construction of Kulkarni et al [1] splits the input points in only one such direction (say horizontally) and then picks a constant number of points in each generated strip. The fact that wedges themselves extend indefinitely in the other (complementary) direction is key to showing that they contain one of the chosen points. In the case of sectors, the additional requirement that they be bounded by a circle, however, makes achieving a similar result considerably harder.

In order to obtain a similar result and at the same time handle the boundedness of our ranges, we employ the fact that all the sectors have a similar radius R. This allows up to further split the points in the perpendicular direction (vertically) but this time, in equally spaced strips of fixed width R. Intuitively, one can think of these vertical strips as bounding the horizontal strips in a way that mimics the way our sectors are bounded wedges. This, however, is not enough to ensure that a good rule exists for picking a constant number of points in each cell. In particular, the rule of Kulkarni et al [1] does not work either. Essentially, this comes from the fact that the intersection of a particular sector with each such cell yields a shape that is rather cumbersome. We deal with this issue by extending the sectors in a way in which each such possible intersection looks roughly like the intersection of an infinite wedge with one of our horizontal strips. In this context, we bear in mind the fact that our sectors are centered at the target and that any point we pick in our ϵ-net represents a sensor that should respect the angle and distance constraints.

To this end, our construction guarantees that we pick sensors that will never violate the angular constraint and will be within $2R$ or $3R$ of the target they are assigned to.

Before we describe our construction, we will briefly outline the one in Kulkarni and Govindarajan [1] and then show how it fails in the case of double-sectors. The first step is to construct an ϵ-net for a more restricted type of wedges called axis-aligned wedges. An *axis-aligned wedge* is a wedge with angle less than $\pi/2$ in which one of the halfspaces is parallel to either the horizontal or the vertical axis. Similarly to Kulkarni and Govindarajan [1], we define an *axis-aligned sector* to be a sector in which one line is parallel to either the horizontal or vertical axis. There are 8 different types of axis-aligned wedges (sectors) and they are denoted being of as Type 1, Type 2 etc. In particular, a Type 1 wedge (sector) is formed by the intersection of a horizontal halfspace and another halfspace defined by a line with a positive slope. [1] focus on constructing an ϵ-net for Type 1 wedges (sectors), noting that the construction can be modified in an intuitive way to work for other types of axis-aligned wedges (sectors).

We now proceed to describe the rule for picking points from Kulkarni et al [1]. Our constructions are similar to theirs, so we briefly summarize the common elements and explain the differences in more detail in each section. The main idea is to divide the input points into $2/\epsilon$ horizontal slices each containing $\epsilon n/2$ points. Each slice is numbered i, $1 \leq i \leq 2/\epsilon$, from bottom to top in terms of the y-coordinate to the highest. For each slice, let P_i be the set of points contained in or above slice i and let H_i denote the convex hull of those points. Kulkarni and Govindarajan [1]



(a) The argument for wedges. (b) The argument for sectors.

Figure 3.4: In the case of wedges, the point $N(p_i)$ must be contained in the wedge. However, that is no longer the case for a sector.

associate with H_i an ordering H_i' of the points on its boundary, in counterclockwise direction starting with the point of highest y-coordinate. For every point $p \in H_i'$, they define $N(p) \in H_i'$ to represent the point right after p in H_i', with $N(p)$ being the first element of H_i' in the case when p is the last element of the ordering. They then consider the restriction of H_i' to slice i and define H_i'' to be the maximal subsequence of H_i' that consists of points on the boundary of H_i that belong to slice i. This set is not empty since it must contain the points of lowest y-coordinate in H_i, which are in slice i. Notice that the points in H_i'' go in counterclockwise direction, essentially from left to right. The rule for picking points in the ϵ-net is the following: for each slice i, pick the point p_i that is the *last* point in H_i'' and its corresponding $N(p_i)$. Notice that the two points essentially correspond to the rightmost vertices in the convex hull whose segment crosses slice i. This leads to an ϵ-net of size $4/\epsilon$. At this point, it is straightforward to see that any ϵ-heavy wedge that fully contains slice i and some other slice above it must either contain p_i or $N(p_i)$ or both. We refer the reader to Fig. 3.4(a) for an intuitive explanation. In addition, Fig. 3.4(b) shows

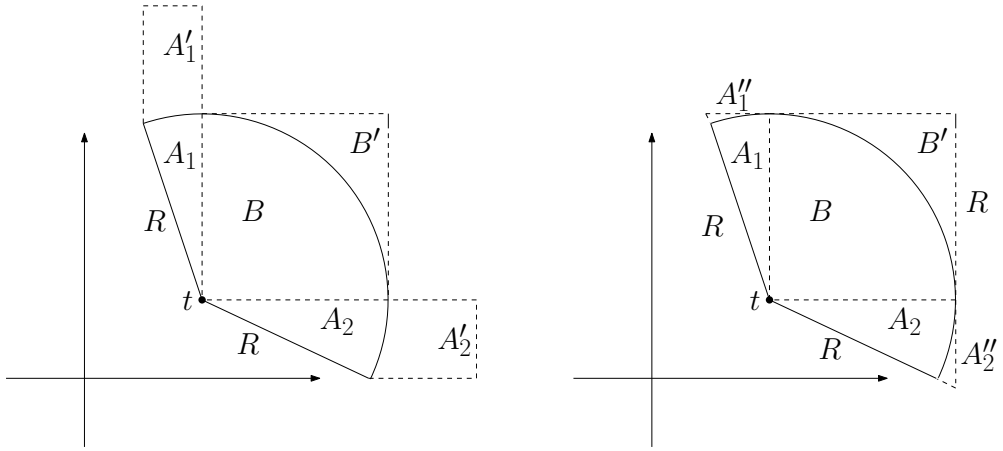
how the argument fails when we consider sectors instead of wedges.

The proof technique for showing that the set is indeed an ϵ-net is to first notice that any wedge that contains $\geq \epsilon n$ points must intersect at least two slices. Look at the first slice i (from bottom to top) that the wedge intersects nontrivially. If the wedge does not already contain p_i, then it must contain $N(p_i)$ because otherwise, the hyperplane defined by the line passing through p_i and $N(p_i)$ and that corresponds to H_i will not contain any point in slice i that is contained in the wedge. We refer the reader to Figure 3.4(a) for a visual representation of this argument. Therefore, such a heavy wedge is guaranteed to contain a point from the ϵ net.

When we deal with sectors instead of wedges, the above argument fails, as seen in Fig 3.4(b). The explanation is that once a wedge intersects a slice, it contains every point in that slice situated to the right of the wedge. That is not true for sector because they can intersect a slice multiple times and therefore contain only a bounded part of that slice. Hence, they do not necessarily have to include $N(p_i)$. In this context, the trick is to divide each slice into bounded *blocks* and pick points from each such block. That will limit how far to the right the sector needs to extend in order to intersect the ϵ-net.

3.4.2 Smaller Hitting Set Constructions for $2R$

We start by considering an arbitrarily chosen system of coordinates and intersecting it with our double-sectors. Each sector then decomposes into at most three separate sectors: two sectors, A_1 and A_2, of acute central angles θ_1 and θ_2 and one



(a) Approximation by right trapezoids and squares.

(b) Approximation by right triangles and squares.

Figure 3.5: Each sector intersects with the coordinate axes and decomposes into at most 3 smaller sectors A_1,A_2 and A_3. We extend each of these smaller sectors into geometric objects for which smaller ϵ-net constructions exist.

sector, B, with central angle $\pi/2$, as seen in Figure 3.5.

We first approximate A_1 and A_2 by two right trapezoids A_1' and A_2' whose bases are parallel to the coordinate axes, as shown in Figure 3.5(a). The larger bases will have a length $R \cdot (1 + \cos\theta_1)$ and $R \cdot (1 + \cos\theta_2)$ respectively. We extend B into a square of side R. This operation enlarges the set of feasible sensor locations in a way that preserves angle coverage but relaxes the distance constraint from R to at most $2R$.

Once we have this new set system, we can modify the construction of Kulkarni and Govindarajan [1] for γ-fat wedges by partitioning the space of input points into horizontal and vertical strips of width R and then constructing an ϵ-net in a similar manner. Since the modification is relatively straightforward, we defer it to later in the section. We obtain an ϵ-net of size $\mathcal{O}(\frac{W}{R} \cdot \frac{1}{\epsilon})$, where $W = \max\{x_r - x_l, y_t - y_b\}$, where x_r and x_l (and y_b and y_l) are the minimum and maximum over all x-

coordinates (y-coordinates) of points in X.

In general, W could be as high as k_{OPT}, so this approximation is only valuable when W is small. In particular, we will use it in the case in which $W = \mathcal{O}(\log\log k_{\mathrm{OPT}})$. When $W = \Omega(\log\log k_{\mathrm{OPT}})$, we employ a different ϵ-net construction. Instead of turning A_1 and A_2 into trapezoids, we turn them into right triangles A_1'' and A_2'' as seen in Figure 3.5(b). We then obtain a set system composed of right triangles and axis parallel rectangles, for which Aronov et al [127] give a randomized $\mathcal{O}(n\log n)$ construction of ϵ-nets of size $\mathcal{O}(\frac{1}{\alpha\epsilon}\cdot\log\log(\frac{1}{\epsilon}))$. Overall, we get that:

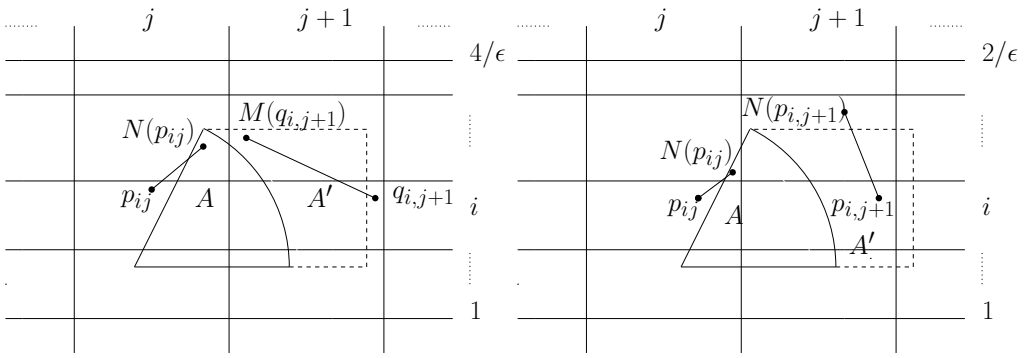
Theorem 3.4. *Given X, T, $\alpha > 0$ and R as above, we can find a set of sensors $S \subseteq X$ such that S $((1-1/\delta)\cdot\alpha)$-covers T within distance $2R$ and $|S| = \mathcal{O}(\frac{1}{\alpha}\log n\log\log k_{\mathrm{OPT}})\cdot k_{\mathrm{OPT}}$, where k_{OPT} is the cardinality of the smallest set of sensors that α-covers T within distance R. The running time of the algorithm is $\mathcal{O}(k_{\mathrm{OPT}}\cdot mn\log m\log n)$, where $n = |T|$ and $m = |X|$.*

Construction of the ϵ-net. Formally, we divide the input points into horizontal slices that each contain $\epsilon n/4$ points each and vertical strips each of width R. The horizontal slices will be numbered i,$1 \le i \le \lceil 4\epsilon\rceil$. The vertical strips will be numbered j, $1 \le j \le \lceil\frac{x_l - x_r}{R}\rceil$, where x_l and x_r represent the leftmost and rightmost x-coordinates of the input points. Each slice i and strip j therefore define the block B_{ij} and let P_{ij} denote the points contained in all the blocks $B_{i'j}$ with $i' \ge i$. In other words, P_{ij} contains all the points in or above slice i restricted to strip j. Let H_{ij} denote the convex hull of the points in P_{ij}, for all $P_{ij} \neq \emptyset$ and, just like before, let H'_{ij} be an ordering of all the points on its boundary, in counterclockwise direction

starting with the topmost point. We keep $N(p)$ as before and consider a new point $M(p) \in H'_{ij}$ to represent the point right before p in the ordering, with $M(p)$ being the first element of H''_{ij} in the case when p is the last element of the ordering. Similarly as before, let $H''ij$ represent the sequence H'_{ij} restricted to block B_{ij}. Notice that H''_{ij} could be empty in the situation in which B_{ij} contains no points. However, we will never deal with such blocks in our proof.

The rule for picking points in the ϵ-net is the following: for each non-empty block B_{ij}, let q_{ij} and p_{ij} be the first and the last points in the ordering H''_{ij} (possibly the same). We then add to the ϵ-net the points $q_{ij}, M(q_{ij}), p_{ij}$ and $N(p_{ij})$. The intuition is that now we want to select essentially the leftmost and rightmost points on the boundary of the convex hull, since our sector will intersect blocks either on the left or on the right. The size of this set is bounded by $4 \cdot \lceil \frac{4}{\epsilon} \rceil \cdot \lceil \frac{x_l - x_r}{R} \rceil$.

Correctness. We will now show that a set constructed in such a way is indeed an ϵ-net for the new set systems we construct. Our strategy will be to extend the sectors in such a way that when we restrict our attention to how they intersect each block, they behave locally as wedges. First, let us notice that a sector with central angle $\geq \pi/2$ can always be decomposed into at most 3 smaller sectors: two axis aligned sectors A_1 and A_2 with acute central angles θ_1 and θ_2 and one sector B with a right central angle and whose radii are axis parallel. Let us consider the sectors described in Figure 3.5(a). For each of the A_1 and A_2 sectors, we extend the axis-parallel radius by $R\cos\theta_1$ and $R\cos\theta_2$ respectively away from the target. This segment will be the larger base of each respective trapezoid. We then draw a line l perpendicular to the extended axis parallel radius. Simultaneously, we attach



(a) Case 1: either the left side of A' must include $N_{p_{ij}}$ or the right side of A' must include $M(q_{i,j+1})$.

(b) Case 2: either the left side of A' must include $N(p_{ij})$ or the right side of A' must include $N(p_{i,j+1})$.

Figure 3.6: Within each block, the trapezoid A' behaves like a wedge.

a segment to the non axis-parallel radius that is parallel to the axis parallel radius. We extend this segment until it intersects l. This point of intersection will define the right trapezoid that encloses the sector. We denote them A'_1 and A'_2. Notice that the maximum distance from t to points in A'_1 is achieved by the top right vertex of A'_1 and is $\sqrt{2R^2(1+\cos^2\theta_1)} \leq 2R$. Similarly for A'_2, the maximum distance from the target is achieved by the bottom right vertex and is $\sqrt{2R^2(1+\cos^2\theta_2)} \leq 2R$. When it comes to B, we draw the square B' of radius R that has the target as its bottom left corner. The maximum distance from the target then becomes $R\sqrt{2}$. We extend all other axis-aligned sectors in a similar way.

Lemma 3.8. *Our construction is an ϵ-net for this modified set system.*

Proof. We focus our attention on right trapezoids that correspond to Type 1 sectors. We will refer to the Type 1 sector as A and to its corresponding trapezoid as A'. Let θ be the central angle of A. Notice that the bottom base of A' has length $(1+\cos\theta)R$ and since it is the longest of the bases, A' will intersect exactly two adjacent vertical strips. We denote those strips as j and $j+1$ and the corresponding

disjoint components of A (and A' respectively) as A_j and A_{j+1}(and A'_j and A'_{j+1} respectively). Notice that since we are considering ϵ-heavy sectors, it follows that either A_j or A_{j+1} must contain at least $\epsilon n/2$ points. By extension, either A'_j or A'_{j+1} must contain at least $\epsilon n/2$ points. We distinguish two types of intersections and show that our construction is an ϵ-net in each case.

Case 1. The vertical line separating strips j and $j+1$ intersect both bases of A', as seen in Figure 3.6(a). In other words, A'_j is another right trapezoid and A'_{j+1} is a rectangle. Suppose A'_j contains at least $\epsilon n/2$ points. Then it means that it must intersect at least 2 horizontal slices. Let j be the minimum index strip that has a non empty intersection with A'_j. Notice that A'_j extends all the way to the right of block B_{ij} and thus, if p_{ij} does not belong to A'_j, then it must be the case that $N_(p_{ij})$ belongs to it. Otherwise, all the points of A_j that are in strip i would not be included in the convex hull H_{ij}. Notice that it does not have to be the case that $N(p_{ij})$ is contained in A_j, but it must be contained in A'_j. In the converse case in which A'_{j+1} contains at least $\epsilon n/2$ points and strip i is identified accordingly, if $q_{i,j+1}$ is not in A_{j+1}, then $M(q_{i,j+1})$ must be, because otherwise all the points that of A_{j+1} that are in strip i will not be included in the convex hull $H_{i,j+1}$. This is the case that motivated us choosing $q_{i,j+1}$ and $M(q_{i,j+1})$: while A'_{j+1} might not extend all the way to the right of $B_{i,j+1}$, it does extend all the way to the left of it.

Case 2. The vertical line separating strips j and $j+1$ intersect only the bottom base of A', as seen in Figure 3.6(b). In other words, A'_j is another right triangle. Suppose A'_j contains at least $\epsilon n/2$ points. Just as before, that means that it must intersect at least 2 horizontal slices. Let j be the minimum index strip that

has a non empty intersection with A'_j. If p_{ij} does not belong to A_j, then it must mean that $N_(p_{ij})$ belongs to it. Otherwise, all the points of A_j that are in strip i would not be included in the convex hull H_{ij}. This is the standard argument used by [1]. In the converse case in which A'_{j+1} contains at least $\epsilon n/2$ points and strip i is identified accordingly, we first note that A'_{j+1} must contain the vertical line that separates strips $j+1$ and $j+2$. In other words, to its right, it contains everything in $B_{i,j+1}$. Thus, one of the points $p_{i,j+1}$ and $N_{p_{i,j+1}}$ must be contained in A'_{j+1}. Notice that this is not true when we consider for A_{j+1}.

We have thus shown that our construction is an ϵ-net for trapezoids that correspond to Type 1 sectors. This construction can be adapted to work for all other types of sectors and their corresponding trapezoids. This provides an ϵ-net of size $\mathcal{O}(\frac{W}{R} \cdot \frac{1}{\epsilon})$ for axis aligned sectors, where $W = \max\{x_r - x_l, y_t - y_b\}$, where x_r and x_l (and y_b and y_l) are the minimum and maximum over all x-coordinates (y-coordinates) of points in X. This transfers into an ϵ-net of size $\mathcal{O}(\frac{W}{R} \cdot \frac{1}{\epsilon})$ for our double-sectors under the relaxed radius constraint. The running time of the algorithm is $\mathcal{O}(k_{\mathrm{OPT}} m \log m)$.

□

Next, we describe how we apply the result of Aronov et al [127]. Apart from extending the right angle sector B into a square B' of side length R, we also extend each sector A_1 and A_2 into right triangles whose sides are parallel to the coordinate axes. Consider an arbitrary sector that is centered at a target t and whose central angle is $\theta \in (0, \pi/2]$, such as the ones shown in Figure 3.5(b) (all other types of

sectors will be similarly modified). We first observe that there is only one way in which we can extend this sector into a triangle without violating the angle constraints. Specifically, we can only extend the two radii of the sector and we will do so by drawing a perpendicular line to the radius that is already parallel to the coordinate axis. The second (non-parallel) radius will be extended until it meets this perpendicular and their intersection will determine the triangle. In this way, we ensure that the radius relaxation is minimized. Specifically, the farthest point from the target will exactly be the new intersection point and will be at a distance of $\frac{R}{\cos\theta}$ from t. Unfortunately, this can be arbitrarily large, since θ can be arbitrarily close to $\pi/2$.

In order to fix this, we rotate the original sector in such a way that we guarantee that both smaller sectors A_1, A_2 have a central angle that is less than $\pi/3$. This way, we can ensure that the maximum distance relaxation is $2R$. As noted before, the central angle of each of the original sectors is in the range $[\pi - \alpha, \pi - \alpha/2]$ and $\alpha \leq \pi/2$. Therefore, if we clockwise rotate the coordinate axes by $i\alpha/6$ radians for $\forall i, i \leq \lceil \frac{12\pi}{\alpha} \rceil$, one of these configurations is going to give us the desired property. We then solve the problem for each of these $\lceil \frac{12\pi}{\alpha} \rceil$ configurations separately.

Assuming that both sectors A_1 and A_2 were successfully extended into the right triangles A_1'' and A_2'', we are now ready to apply the results of Aronov et al [127]:

- for the set system formed out of the squares B, notice that they are axis-parallel, and we get an ϵ-net of size $\mathcal{O}(\frac{1}{\epsilon} \cdot \log\log(\frac{1}{\epsilon}))$.

- for the set system composed of right triangles whose sides are parallel to the coordinate axes, as noted by Aronov et al [127], we can assume that each triangle has the same orientation: the perpendicular sides of each triangle meet at the lower left vertex of the triangle (the semi-canonical right triangle considered by Aronov et al [127]). We can then directly apply their results and obtain an ϵ net of size $\mathcal{O}(\frac{1}{\epsilon} \cdot \log\log\epsilon)$. Notice that as such, the size of the ϵ-net does not depend on α. By computing such ϵ-nets for each of the $\lceil\frac{12\pi}{\alpha}\rceil$ configurations, we obtain a total set of size $\mathcal{O}(\frac{1}{\alpha\epsilon} \cdot \log\log(\frac{1}{\epsilon}))$.

In total, we obtain an ϵ net for our overall set system of size $\mathcal{O}(\frac{1}{\alpha\epsilon} \cdot \log\log(\frac{1}{\epsilon}))$. By decomposing each double-sector into sectors and then extending those into their appropriate shapes, we obtain a similarly sized ϵ-net for our double-sectors with the relaxed radius constraint. The expected running time of the algorithm is $\mathcal{O}(n \log n)$ [127].

Combining these two approaches give us the results in Theorem 3.5.

3.4.3 Smaller Hitting Set Constructions for $3R$

In an attempt to reduce the approximation factor in this latter case, we consider relaxing the distance constraint and allowing the chosen sensors to be within distance $3R$ of the targets. In other words, we extend the radius of our sectors from R to $3R$. Inspired by the construction of Kulkarni and Govindarajan [1], we then propose a deterministic rule for picking sensors and obtain a "relaxed" ϵ-net of size $\mathcal{O}(\frac{R_I}{R} \cdot \frac{1}{\epsilon})$, where R_I is the diameter of the largest enclosing ball of all possible

sensor locations. We note that this construction is not cyclical: we are not building an ϵ-net for sectors of radius $3R$. In our case, our heavy ranges have the special property that at least $\epsilon \cdot n$ of their points are actually contained inside the sector of radius of R. The main difference between our construction and the one in [1] is the fact that the objects the latter considers are infinite and, as such, allow for simpler grid-based constructions. In fact, this distinction is indeed the source of the additional $\frac{R_I}{R}$ factor that we incur in our bound. To our knowledge, this is the first ϵ-net construction whose size depends linearly on $\frac{1}{\epsilon}$ and the ratio of the diameter of the input space to the size of the ranges (that is, when size can be appropriately defined). We note that the $\mathcal{O}(\frac{1}{\epsilon})$ construction of Pach and Woeginger [**?**] for translates of convex polygons does implicitly depend on solving the problem for points contained inside a bounded square. It is unclear, however, how to adapt their method for the case in which the ranges are sectors of similar radius but can have arbitrary central angles and orientations.

In general, R_I could be as high as k_{OPT}, so in order to get rid of this dependency, we employ the shitting strategy of Hochbaum and Maass [128] that allows us to construct a global solution by individually solving the problem on instances of fixed width. This further incurs a constant factor in our final solution size. The analysis then yields an overall hitting set of size $\mathcal{O}(k_{\mathrm{OPT}})$ that achieves the desired $(\alpha - \epsilon)$-coverage and is within distance $3R$ of the targets. Formally, we get that:

Theorem 3.5. *Given X, T, $\alpha > 0$ and R as above, we can find a set of sensors $S \subseteq X$ such that S $((1 - 1/\delta) \cdot \alpha)$-covers T within distance at most $3R$ and $|S| =$*

$\mathcal{O}(\log \delta) \cdot k_{\mathrm{OPT}}$. *The running time of the algorithm is* $\mathcal{O}(k_{\mathrm{OPT}} \cdot mn \log m \log n)$.

We now proceed to formally describe the ingredients. As usual, we will focus on describing an ϵ-net for sectors, noting that this translates into a 2ϵ-net for double sectors. We start by designing an ϵ-net for extended sectors of radius $3R$. In other words, if A is an ϵ-heavy sector of radius R with central angle $\theta \in [\pi - \alpha, \pi - \alpha/2]$ and A' is the corresponding sector we obtain by extending the radius of A to $3R$, then we must have that the ϵ-net H we construct intersects A' non trivially.

Theorem 3.6. *In the case in which the ranges are induced by sectors of radius R that have central angle in the range $[\pi - \alpha, \pi - \alpha/2]$, for $\alpha \in [0, \pi/2]$, there exists an ϵ-net construction H of size $\mathcal{O}(\frac{R_I}{R} \cdot \frac{1}{\epsilon})$ that guarantees that, for each ϵ-heavy sector, there exists a point in H that intersects the corresponding extended sector of radius $3R$. In this result, R_I represents the radius of the smallest enclosing ball of all the input points.*

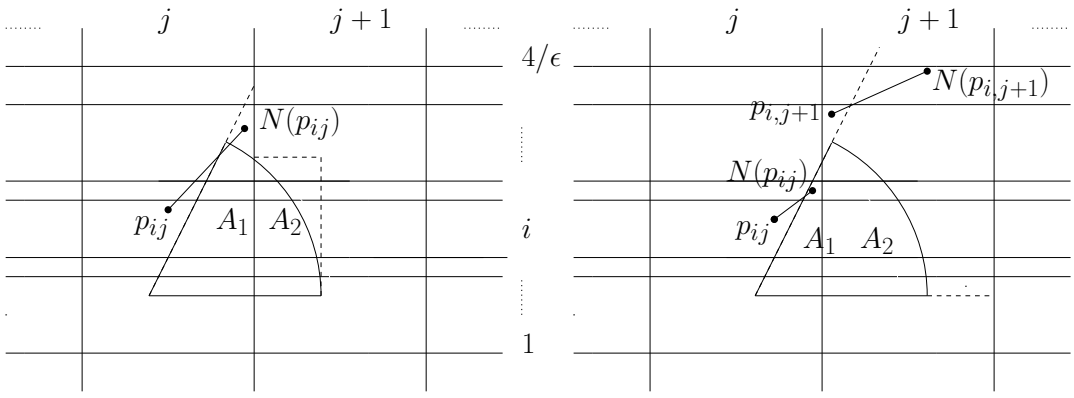
Proof. Consider an arbitrary system of coordinate axes. We first decompose each sector A into smaller ones that have one side parallel to the coordinate axes, which we call *axis parallel sectors*. Because its central angle is smaller than π, A will decompose into at most 3 smaller axis parallel sectors, each with central angle at most $\pi/2$. If we build an ϵ-net for axis parallel sectors, this will turn into a 3ϵ-net for the more general type of sectors, so we will restrict out attention to the former. Notice that there are 8 different types of axis parallel sectors. We will describe the construction for one such type and note that it can be intuitively modified to work for the other types. In particular, we will look at sectors that are formed by

the intersection of one horizontal halfspace and another one defined by a line with positive slope, as seen in Fig. 3.4(a).

For reasons that will become evident later in the proof, however, we must require that all possible axis-parallel sectors either have central angle of at most $\pi/3$ or exactly $\pi/2$. To that end, we consider 2 additional coordinate axes that are rotated copies of the initial coordinate axes, each by a factor of $\pi/3$, one in clockwise and the other one in counterclockwise direction. It can be easily checked that now, given any sector A, there exists a system of coordinate axes that will decompose A into at most 3 axis parallel sectors that all have central angles that are either $\leq \pi/3$ or exactly $\pi/2$. By constructing separate ϵ-nets for each such coordinate axes, we incur an additional factor of 3 in our solution size.

Construction of the ϵ-net. We now proceed to describe the rule for picking points. In this part, our construction is similar to the one by Kulkarni et al [1]. We modify it in the following ways: first, we will have $4/\epsilon$ horizontal slices each containing $\epsilon n/4$ points each. We will further divide the input space into *vertical strips* of width R each. The horizontal slices will be numbered i,$1 \leq i \leq \lceil 4\epsilon \rceil$. The vertical strips will be numbered j, $1 \leq j \leq \lceil \frac{x_l - x_r}{R} \rceil$, where x_l and x_r represent the leftmost and rightmost x-coordinates of the input points. Each slice i and strip j therefore define the block B_{ij} and let P_{ij} denote the points contained in all the blocks $B_{i'j}$ with $i' \geq i$. In other words, P_{ij} contains all the points in or above slice i restricted to strip j.

Let H_{ij} denote the convex hull of the points in P_{ij}, for all $P_{ij} \neq \emptyset$ and, just like before, let H'_{ij} be an ordering of all the points on its boundary, in counterclockwise



(a) Case 1. (b) Case 2.

Figure 3.7: Within each block, A behaves almost like a wedge.

direction starting with the topmost point. Similarly, let H''_{ij} represent the sequence H'_{ij} restricted to block B_{ij}. Notice that H''_{ij} could be empty in the situation in which B_{ij} contains no points. However, we will never deal with such blocks in our proof. Our rule for picking points will still pick p_{ij} and $N(p_{ij})$, where p_{ij} is the last element in H''_{ij} and $N(p_{ij})$ is defined as before. In addition, we also pick the leftmost and rightmost points in each block B_{ij}. The size of the final set H will be $4 \cdot \lceil \frac{4}{\epsilon} \rceil \cdot \lceil \frac{x_l - x_r}{R} \rceil$.

Correctness. We will now consider an ϵ-heavy axis parallel sector A and show that H will intersect A's extended sector of radius $3R$. We will focus on the case in which the central angle θ of A is $\leq \pi/3$. We first begin by noticing that, since A is a sector of radius R, it intersects at most 2 consecutive horizontal strips. In particular, we will have one of two cases, as seen in Fig. 3.7.

Case 1. The vertical line separating strips j and $j+1$ intersects the arc of A, as seen in Fig. 3.7(a). In this case, A decomposes into a left component A_1 and a right component A_2. One of these two must contain at least $\epsilon n/2$ points. Suppose it is the left component A_1. Since each horizontal slice contains exactly $\epsilon n/4$ points,

A_1 must intersect at least 2 horizontal slices (not necessarily consecutive). Let j be the minimum index strip that has a non empty intersection with A_1. If p_{ij} is contained in A_1, we are in good shape. Otherwise, $N(p_{ij})$ must be contained in the space we get by extending the non horizontal side of A until it meets the closest vertical strip to the right. This is because, otherwise, th convex hull H_i would not cover the points in A_1 that are contained in slice i and all the other slices that are above i (of which there is at least one). The farthest possible location for $N(p_{ij})$ is exactly the intersection point with the vertical line. In this situation, the distance from the target is at most $R/\cos\theta$) and since $\theta \leq \pi/3$, we get that the maximum distance to the target is $2R$.

When the right component A_2 has more than $\epsilon n/2$ points, we note that it must also intersect at least 2 horizontal slices. In this case, we consider the leftmost point in slice i. It must be contained in the smallest axis parallel rectangle that encloses A_2. The farthest point in this scenario is the upper right corner, which is at most $R\sqrt{1+\sin^2\theta} \leq R\sqrt{2}$ away from the target.

Case 2. The vertical line separating strips j and $j+1$ intersect a side of the sector, as seen in Fig. 3.7(b). We again decompose into the right and left components A_1 and A_2 respectively. If A_1 contains more than $\epsilon n/2$ points, we are guaranteed that either p_{ij} of $N_{p_{ij}}$ are contained in it, by a similar argument as above. If, on the other hand, A_2 contains more than $\epsilon n/2$ points, in the case in which $p_{i,j+1}$ is not contained in it, then $N(p_{i,j+1})$ will be contained in the extended object that we get from intersecting the non-parallel side of A with the closest vertical line to the right. The farthest point in this object is at at most $R(1+1/\cos\theta) \leq 3R$ away

from the target.

The other case we need to consider is when A has a central angle of $\pi/2$. In that case, we extend it to the smallest enclosing square (of side length R). When the square intersects the vertical line between strips j and $j+1$, separating the square into A_1 and A_2, we will have that either A_1 will contain the rightmost point in box B_{ij} or A_2 will contain the leftmost point in $B_{i,j+1}$. The farthest point from the target is the top right corner of the square and is $R\sqrt{2}$ away from the target.

We have therefore shown that our construction will always contain points that are close to the targets that correspond to the heavy sectors. Additionally, we must note that all of these points are not just a distance at most $3R$ away from the target: they are also contained in A's extended sector of radius $3R$. In other words, we are guaranteed that the initial angle constraint that A imposes is not violated.

Finally, we must note that, in this specific construction, we consider the horizontal width $W_x = x_r - x_l$ because we are dealing with a specific kind of axis parallel sectors given by a specific choice for our coordinate axes. For the other types of axis parallel sectors, the factor will depend on $W_y = y_t - y_b$, where y_t and y_b are the y-coordinates of the top and bottom vertices in the input space. We get that, for a fixed coordinate system, the size of the ϵ-net will be at most $\mathcal{O}(1) \cdot \lceil \frac{1}{\epsilon} \rceil \cdot \lceil \frac{W}{R} \rceil$, where $W = \max\{W_x, W_y\}$. Combining this with the other constructions that correspond to the rotated systems of coordinate axes, we get that the size of the ϵ-net will be at most $\mathcal{O}(1) \cdot \lceil \frac{1}{\epsilon} \rceil \cdot \lceil \frac{R_I}{R} \rceil$, where R_I is the diameter of the smallest enclosing ball of all the input points.

□

Remark 1. An alternative way of thinking about the above construction is that by relaxing the radius constraint to $3R$, we can include our double sectors in slightly larger geometric objects for which we construct an ϵ-net of size $\mathcal{O}(\frac{R_I}{R} \cdot \frac{1}{\epsilon})$. The way we extend each sector is deterministic and depends on the choice of coordinate axes and the corresponding type of axis-parallel sector. We note, however, that this is not the same as constructing an ϵ-net for the extended sectors of radius $3R$. Each of our objects are subsets of the extended sector of radius $3R$ but their construction relies crucially on the fact that we were extending from sectors of radius R.

Remark 2. We note that our choice for the threshold of $\pi/3$ for the central angle of an axis parallel sector was rather arbitrary. As we have seen, it was used in bounding the distance from the farthest point of the ϵ-net guaranteed to be in the extended sectors to the target at the center of the sector. Smaller distance guarantees can be obtained by considering smaller thresholds at the expense of requiring more rotated copies of systems of coordinate axes (which in turn will increase the approximation factor in the overall ϵ-net size).

We will now show how to use this construction in order to build a set S' that is within a constant of the size of the optimal hitting set and guarantees that the points we pick are contained in the extended sectors of radius $3R$. Specifically, we will show the following:

Lemma 3.9. *Let $\epsilon > 0$ be such that $\alpha - \epsilon \leq \pi/3$ and $\epsilon \leq \alpha/2$. Let $S \subseteq X$ be a set that $(\alpha - 2\epsilon)$- covers T within distance $R' \geq 0$. Then we can find a set $S' \subseteq X$ such that S' $(\alpha - \epsilon)$-covers T within distance $\max\{3R, R'\}$ and $|S'| = |S| + \mathcal{O}(1) \cdot k_{\mathrm{OPT}}$.*

Proof. As we have seen before, we can construct S' as a hitting set for a new set of geometric objects that we obtained by extending each double sector. Specifically, given the set system $\mathcal{F}'(X, \mathcal{R}')$ containing sectors of radius R, we constructed a new set system $\mathcal{F}'_{\text{new}}(X, \mathcal{R}'_{\text{new}})$ for which we gave an ϵ-net construction of size $\mathcal{O}(\frac{R_I}{R} \cdot \frac{1}{\epsilon})$. First notice that the size τ^*_{new} of the optimal hitting set for $\mathcal{F}'_{\text{new}}$ is upper bounded by the size τ^* of the optimal hitting set for $\mathcal{F}'$. A straightforward application of Brönnimann and Goodrich [109] would therefore get a set S' with $|S'| = \mathcal{O}(\frac{R_I}{R} \cdot \tau^*)$.

In order to get a constant approximation factor instead, we will employ the shifting technique of Hochbaum and Maass [**?**] as it applies to our case. Before we describe its application, however, we first note that we will construct S' by first restricting ourselves to each of the systems of coordinate axes that we considered in the previous construction. Each sector in $\mathcal{F}'$ corresponds to exactly one coordinate system, specifically the one in which it has the desired property that it decomposes into at most 3 smaller sectors that have a central angle that is either $\leq \pi/3$ or exactly $\pi/2$. Each coordinate system i therefore corresponds to a different set of ranges $\mathcal{F}'_{\text{new},i}(X, \mathcal{R}'_{\text{new},i})$ that are extensions of the sectors of radius R affiliated with that coordinate system. In other words,

$$\mathcal{R}'_{\text{new}} = \mathcal{R}'_{\text{new},1} \cup \mathcal{R}'_{\text{new},2} \cup \mathcal{R}'_{\text{new},3}.$$

As such, we will construct S' by actually constructing three separate hitting sets which we will denote as S'_1, S'_2 and S'_3 and then taking their union. Let $\tau^*_{\text{new},1}$, $\tau^*_{\text{new},2}$ and $\tau^*_{\text{new},3}$ be the size of the optimal hitting set in each of these set systems. It follows that $\tau^*_{\text{new},1} \leq \tau^*_{\text{new}}$ etc. We will now construct hitting sets such that each

$|S'_i|$ is within a constant fraction of $\tau^*_{\text{new},i}$.

We now fix a coordinate system and notice that the appropriate ϵ-net construction from before has size $\mathcal{O}(1) \cdot \lceil \frac{1}{\epsilon} \rceil \cdot \lceil \frac{W}{R} \rceil$, where $W = \max\{W_x, W_y\}$. In other words, for a set of points and ranges that are all contained inside a space with $W \leq k \cdot R$, we would get a hitting set of size at most $\mathcal{O}(k)$ times the size of the optimal hitting set for those points. We will denote this local algorithm as A.

The shifting technique of Hochbaum and Maass [128] provides a way of analyzing the global approximation factor we would get by applying algorithm A on instances of bounded W and returning the union of all the hitting sets we compute. We will not reproduce the argument entirely in this paper, only mention how it applies to our case. We first need to define the appropriate partition into smaller instances. Suppose we first divide X into vertical strips of width $l \cdot 6R$, where $l \geq 2$ is an arbitrary parameter. We now need to decide how we will appropriately partition the ranges in $\mathcal{R}'_{\text{new},i}$. If a range is completely contained in a strip, we assign it to it. Otherwise, assign the range to the strip that contains its center (the target that is the center of the sector). Each such range is contained in a double sector of radius $3R$ which can intersect at most one additional strip. This is what motivates us to consider strips of size $l \cdot 6R$ instead of $l \cdot 2R$. For each such strip, we employ an algorithm A' (to be defined later) that will take as input the ranges associated with that strip and the set of sensors contained in the enlarged strip of side length $(l+1) \cdot 3R$ (i.e. the original strip padded with $3R$ on the left and right).

Now we can see how the global optimal hitting set behaves with respect to the localized optimal hitting sets in each of the strips. We notice that a point in

the global optimum can appear at most twice in the induced hitting sets for two consecutive strips. Moreover, when we consider the other possible partitions into strips of width $l\dot{6}R$ (obtained by shifting a strip to the right by $6R$), the sets of points that get double counted in each of the partitions are disjoint. This is because, if a point in the optimal global hitting set gets double counted in one partition, it will never get double counted again in any of the shifted $l-1$ partitions since the ranges that contribute to it being double counted are no longer in two distinct strips. Therefore, the analysis of Hochbaum and Maass [128] applies and we get that, if the algorithm A' solves the hitting set problem for its input strip within a factor of α_l, we are guaranteed to obtain an overall approximation guarantee of $\alpha_l \cdot (1 + \frac{1}{l})$ (by taking the minimum hitting set over all possible l partitions). The way we are going to determine α_l is by reproducing the above argument and splitting each horizontal strip of width $(l+1) \cdot 6R$ into vertical strips of length $l \cdot 6R$. At this point, we can use algorithm A and give to it a set of points and ranges that are fully contained in a square of side length $(l+1) \cdot 6R$. At this point, A will return a hitting set that is within a factor of $\mathcal{O}(l)$ of the optimum. This, in turn, will mean that the approximation factor for A', α_l is also at most $\mathcal{O}(l)$. By choosing $l = 2$, we get a constant factor of approximation overall. The actual constants depend directly on the ϵ-net constructions used and an improvement in that direction would directly translate in an improvement for the hitting set algorithm.

□

3.5 Euclidean Fault Tolerant k-suppliers

Another interesting special case of (α, R)-AngDist is the one in which $\alpha = 0$, since it requires us to place two distinct sensors within distance R of each target, in the spirit of fault- tolerance. In this context, a slightly different but relevant problem is the FAULT TOLERANT $-$ k $-$ SUPPLIERS problem, as defined by Khuller et al. [19]. Before we define the FAULT TOLERANT $-$ k $-$ SUPPLIERS problem, we would like to first talk about the (k, r)-CENTER problem as defined by Bar-Iral et al [129], since it captures nicely the bi-criteria nature of our problem. In this problem, we are given a set of n input points and the goal is to choose k points (centers) out of the input points such that every point is within distance r of at least some chosen center. The underlying structure is a graph and the distance is a given function on pairs of vertices (metric or not). There are two ways to attack this problem, each of which gives rise to different optimization problems. On one hand, one could fix the number of centers to be k and then focus on minimizing r: this is the well-known k-CENTER problem. On the other hand, one could fix the radius r and consider minimizing k: this then becomes the r-DOMINATION problem [129].

The (k, r)-SUPPLIERS problem is similar to (k, r)-CENTER. The main difference is that the set of centers must be picked from a separate set. Formally, we are given a bipartite graph $G = (U, V, E)$ in which U is the set of suppliers and V is the set of clients, and a distance function $d : U \times V \rightarrow \mathbb{R}^{\geq 0}$. In the k-SUPPLIERS problem, the objective is to find a subset $S \subseteq U$ of suppliers of cardinality k such that all the

clients in V are within radius r of a center. Specifically, we desire a set S that satisfies $\min_{S\subseteq U,|S|\leq k} \max_{v\in V} \min_{u\ \in S} d(v,u)$. The analogue SUPPLIERS r-DOMINATION problem requires a set S of minimum size such that $\max_{v\in V} \min_{u\ \in S} d(v,u) \leq r$. An interesting variation of the (k,r)-SUPPLIERS is the FAULT TOLERANT (k,r)-SUPPLIERS problem, in which we require a client to be close to at least δ of the suppliers, for a given parameter $\delta \leq k$. Specifically we define

$$d^{(\delta)}(v,S) = \min_{A\subseteq S,|A|=\delta} \max_{u\in A} d(v,u).$$

Then the δ-NEIGHBORS k-SUPPLIERS problem (or alternatively, the FAULT TOLERANT$-$ $k-$ SUPPLIERS problem with parameter δ) requires us to find a set $S \subseteq U$ of cardinality k such that S satisfies $\min_{S\subseteq U,|S|\leq k} \max_{v\in V} d^{(\delta)}(v,S)$. Conversely, the δ-NEIGHBORS SUPPLIERS r-DOMINATION problem requires us to find a set $S \subseteq U$ of minimum cardinality such that

$$\max_{v\in V} d^{(\delta)}(v,S) \leq r.$$

Then the δ-NEIGHBORS SUPPLIERS r-DOMINATION problem with $\delta = 2$ and $r = R$ is exactly (α, R)-ANGDIST with $\alpha = 0$.

In this section, we outline a general algorithm for the case in which the input points are in the $\mathbb{R}^d$, the distance function is the l_2 metric, and δ is arbitrary. Specifically, we obtain the following:

Theorem 3.7. *There exists a polynomial time $(1+\sqrt{3})$-approximation algorithm for the Euclidean δ-NEIGHBORS k-SUPPLIERS problem in any dimension.*

While the k-SUPPLIERS problem requires us to minimize the covering radius

rather than the size of the set of centers, the analysis of the above algorithm actually relies on solving the SUPPLIERS r-DOMINATION problem when r is relaxed. Specifically, let r^* be the optimal radius for the k-SUPPLIERS problem. The algorithm uses a standard technique for dealing with bottleneck problems introduced by Hochbaum and Shmoys [93]: the optimal distance must be one of the pairwise distances between suppliers and clients and thus, we can find it by doing binary search. Our algorithm hence works with a guess R for r^* and computes a solution to the SUPPLIERS r-DOMINATION with $r = (1 + \sqrt{3}) \cdot R$. We therefore get the following result as well:

Theorem 3.8. *Given X, T, and R as above, we can find a set of sensors $S \subseteq X$ such that S 0-covers T within distance $R \cdot (1 + \sqrt{3})$ and $|S| = k_{\text{OPT}}$, where k_{OPT} is the cardinality of the smallest set of sensors that 0-covers T within distance R. The running time of the algorithm is $\mathcal{O}(n^2 \log n(m + n \log n))$, where $n = |T|$ and $m = |X|$.*

In the $(1 + \sqrt{3})$-approximation for k-suppliers, the key observation made by Nagarajan et al [118] is the following:

Lemma 3.10. *(Nagarajan et al [118]) If three clients have pairwise distances strictly greater than $\sqrt{3} \cdot R$, then they cannot be covered within distance R by the same supplier.*

Proof. We restrict our attention to points in $\mathbb{R}^2$, mentioning that they also follow for $\mathbb{R}^d$. We refer the reader to Figure 3.8 for an intuitive explanation. Let t_1, t_2, t_3 be the three clients and suppose by contradiction that there exists a supplier s such

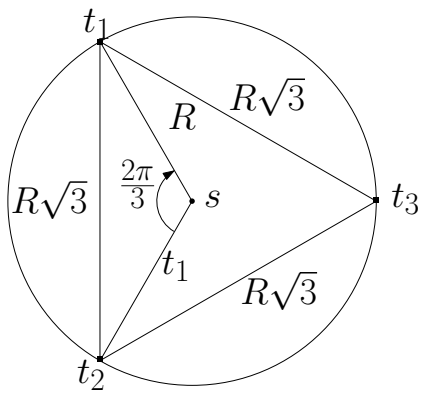


Figure 3.8: Suppose you are given three clients t_1,t_2,t_3 all within pairwise distance exactly $R\sqrt{3}$ of each other and within distance exactly R of a supplier s. The angle $\angle t_1 s t_2$ then becomes $2\pi/3$ and similarly for $\angle t_2 s t_3$ and $\angle t_3 s t_1$. However, once we require that $d(t_1, t_2) > R\sqrt{3}$, we must have that $\angle t_1 s t_2 > 2\pi/3$. Moreover, the same holds true when $d(s, t_1)$ and $d(s, t_2)$ are $\leq R$.

that $d(s, t_1) \leq R, d(s, t_2) \leq R$ and $d(s, t_3) \leq R$. Restrict your attention to the triangle $\triangle t_1 s t_2$. Because $d(t_1, t_2) > r\sqrt{3}$, that implies that $\angle t_1 s t_2 > 2\pi/3$. In a similar way, we have that $\angle t_1 s t_3 > 2\pi/3$ and $\angle t_2 s t_3 > 2\pi/3$. But we have that

$$\angle t_1 s t_2 + \angle t_2 s t_3 + \angle t_3 s t_1 = 2\pi,$$

which leads to a contradiction. □

Notice that this is a more advanced observation that the classical one we have mentioned before. In that observation, only two far apart targets were considered at each time (i.e. with distance greater than $2R$) with the guarantee that they cannot share the same vertex. In the Euclidean context, this observation can be extended to three far apart targets as long as we require that the pairwise distance between them is greater than $\sqrt{3}R$. This is where the particular structure of the Euclidean space comes into place, since, in general metric spaces, this observation does not hold true.

The authors hence restrict their attention to a maximal set P of clients that have pairwise distances $> \sqrt{3} \cdot R$. Notice that this is a more advanced observation that the classical one we mention in the introduction. In that observation, only two far apart targets were considered at each time (i.e. with distance greater than $2R$) with the guarantee that they cannot share the same vertex. In the Euclidean context, this observation can be extended to three far apart targets as long as we require that the pairwise distance between them is greater than $\sqrt{3}R$. This is where the particular structure of the Euclidean space comes into place, since, in general metric spaces, this observation does not hold true.

They construct the graph G that has P as the vertex set and an edge between two clients in P if they can be covered within distance R by the same supplier. Since any supplier can serve at most two clients in P, there exists a one-to-one correspondence between suppliers and edges in G and the problem of clustering the clients in P becomes equivalent to finding an minimum size *edge cover* of G. An edge cover of G is a subset of edges such that each vertex of G is incident to one of the selected edge. In other words, an edge cover would correspond to a set of sensors that cover each of the targets in P within distance R. This problem can be solved optimally in polynomial time [130]. Once they compute a set $S \subseteq X$ of suppliers that cover the clients in P within distance R, all the other clients are within distance $\leq \sqrt{3}$ away from P and hence, within distance $\leq (1 + \sqrt{3})$ away from S.

In the case of FAULT TOLERANT k-SUPPLIERS, the same observation applies and hence, we still have a one-to-one correspondence between edge in G and suppliers

in X. The structure of the optimal solution, however, is different. It corresponds to a SIMPLE b-EDGE COVER: a subset of edges such that each vertex $v \in P$ is incident on at least b_v edges, for all $v \in P$. In our case, $b_v = \delta$. It is known that computing a minimum size SIMPLE b-EDGE COVER problem can be done in polynomial time by computing a maximum size b-MATCHING [130]. The latter can be solved in time $\mathcal{O}(n^2 \log n(m + n \log n))$, where n is the number of vertices and m is the number of edges [131]. The analysis that obtains a $(1 + \sqrt{3})$-approximation remains the same.

3.6 Complexity of Angular Constraints

In this chapter, we address the theoretical complexities of the problems introduced when we would like to satisfy the angle constraints optimally. First, we discuss the SET COVER approach we outlined in the introduction and how the geometry of the angle constraints can be used to obtain a $\mathcal{O}(k_{\text{OPT}} \log k_{\text{OPT}})$-approximation for α-ANG, (α, R)-ANGDIST and α-ANGART (Section 3.6.1).

In terms of the complexity of the problems introduced, notice that, while the (α, R)-ANGDIST and the α-ANGART problems are similar to the r-DOMINATION and the ART GALLERY problems, they are not quite generalizations of them. The case in which $\alpha = 0$ essentially imposes no angle constrains on the relative position of the chosen sensors but it does require that two distinct sensors be assigned to each target. An optimal solution in this case would be a solution for the r-DOMINATION and the ART GALLERY problems, but we cannot guarantee an equivalence between solution sizes. In particular, given a solution to the latter problems, it is

not clear how many new sensors we would have to add in order to get a solution for (α, R)-AngDist or α-AngArt. We nevertheless believe these problems to be NP-complete and to that end, we present some partial progress.

In particular, we generalize the pairwise dependencies between sensors to a more general framework which we call Pairwise Selection that we show is a generalization of the Min Rep problem introduced by Kortsarz [122]. As such, it is hard to approximate within a factor better than $2^{\log^{1-\epsilon} n}$, for any $0 < \epsilon < 1$ under some standard complexity assumptions (Section 3.6.2). It is worthwhile to note that Min Rep is the minimization version of the established Label Cover problem introduced by Arora et al [132]. The latter is a canonical problem used in showing strong hardness of approximation results for a lot of problems, as observed by Arora and Lund in [133]. We believe it to be rather interesting that Pairwise Selection, which is mainly motivated by its applicability, generalizes a problem that is normally employed for its theoretical versatility.s

Having said that, these above results do not necessarily hold when the underlying space is Euclidean and the constraints are geometric. To this end, we make partial progress by exhibiting a reduction from dominating set that might be used in proving NP-completeness for α-Ang, (α, R)-AngDist and α-AngArt (Section 3.6.3).

3.6.1 Set Cover and the Geometry of the Sets

In the introduction, we considered α-Ang as an instance of Set Cover: let the universe of elements be the set of targets T and for each pair of sensors $s, s' \in X$, let $S_{(s,s')}$ be the set of targets that (s, s') α-covers. We adapt the generic Set Cover algorithm for the case in which we want to minimize the number of sensors that cover the universe. Let S_{OPT} be the size of the smallest set of sensors that provide α-coverage and let k_{OPT} be its size. We use two greedy heuristics:

- Pick-pair: at each step, pick a fresh pair $p^{\text{new}} = \{s_1^{\text{new}}, s_2^{\text{new}}\}$ that maximizes the number of targets it covers from the ones that are left unconvered
- Pick-sensor: pick a fresh sensor s^{new} that maximizes the number of targets that are covered by pairs including s^{new} and the sensors already picked.

Lemma 3.11. *The above heuristics gives a solution that uses at most* $2\binom{k_{\text{OPT}}}{2}(1 + \log(\frac{n}{\binom{k_{\text{OPT}}}{2}}))$ *number of sensors.*

Proof. In the initial round, we can choose the pair that maximizes the number of targets covered. Since $\binom{S_{\text{OPT}}}{2}$ covers the universe, it follows that there must exist a pair of sensors $s_1, s_2 \in S_{\text{OPT}}$ that covers at least $\frac{n}{\binom{k_{\text{OPT}}}{2}}$ targets. For ease of calculation, let us denote $\gamma = \binom{k_{\text{OPT}}}{2}$.

By the properties above, we can guarantee that the pair we pick covers at least $\frac{1}{\gamma}n$ of the targets. So now we are left with covering at most $n(1 - 1/\gamma)$ targets. S^* guarantees that there exists a pair that covers at least a fraction $1/\gamma$ of the remaining targets. However, if we use Pick-pair and pick a new pair, we cannot guarantee that

the we will pick the maximum coverage pair of the remaining ones. If, however, we use Pick-sensor and pick the sensor that maximizes coverage with the newly formed pairs, we also cannot guarantee that it achieves the maximum. Instead, we can try both approaches and look at the disjoint pairs (as in Pick-pair) and the ones that are formed with the already chose sensors (as in Pick-sensor) and pick the one with the best coverage. Then we can indeed guarantee that its coverage is at least $1/\gamma$ of the remaining nodes.

At this point, we can guarantee that we are left with at most $n(1-1/\gamma)^2$ uncovered targets. Notice that adding a new pair or a new target to the set can actually cover much more than a fraction of $1/\gamma$ but this is not guaranteed. Notice also that, in each step, we add at most two new sensors. If we repeat this argument for $t = \gamma \log n$ steps, we are left with no uncovered vertices and our solution size is at most $2\gamma \log n$. We therefore get a $((k_{\mathrm{OPT}} - 1)\log n)$ -approximation for the problem of picking a minimum number of sensors that ensure α-coverage.

This can be further improved by noticing that, once we have $\leq \gamma$ targets left, we can cover them by picking at most 2γ pairs, since each new pair must cover at least one target. That gives us that the number of rounds is $\gamma \log(n/\gamma)$ and our total solution size is at most:

$$2\gamma \log(n/\gamma) + 2\gamma = 2\tbinom{k_{\mathrm{OPT}}}{2}(1 + \log(\tfrac{n}{\binom{k_{\mathrm{OPT}}}{2}})).$$

This in turn leads to an approximation factor of $(k_{\mathrm{OPT}} - 1)(1 + \log(\frac{n}{\binom{k_{\mathrm{OPT}}}{2}})$. □

One can construct examples for which this analysis is tight. Other adaptations of greedy algorithms for Set Cover run into the same problem: we cannot guar-

antee a better upper bound on the size of the optimal set cover. Moreover, a better upper bound would still have to depend on k_{OPT} and $logn$. Notice that, in general, a solution of size k for the Set Cover instance requires a number of sensors that is greater than $\sqrt{k}$ but smaller than $2k$. Let S^* be the optimal solution to the Set Cover instance and k^* its size. We get that $k^* \leq k_{\text{OPT}}^2$ and that the number of sensors that S^* uses is at least k_{OPT}. Therefore, a solution of size $k = k^* \log n$ would use at least $\sqrt{k^* \log n}$ sensors and at most $2k^* \log n$ sensors. This becomes a solution that uses at least $\sqrt{k_{\text{OPT}} \cdot \log n}$ sensors and at most $\log n \cdot k_{\text{OPT}}^2$ sensors.

One way in which we can improve the above approximation factor is by exploiting the underlying geometry: each $S_{(s,s')}$ is induced by the symmetric difference of two disks $\mathcal{D}_1$ and $\mathcal{D}_2$ determined by s and s', as shown in Figure 3.11 and formally proven below. Notice that this set system can be obtained from the set system made of sensors and their corresponding $\mathcal{D}_1, \mathcal{D}_2$ by applying the operation of set union, intersection and then difference to the disks $\mathcal{D}_1, \mathcal{D}_2$. Since the set system made of points and disks has finite VC-dimension, it follows that the former set system also has finite VC-dimension [123].

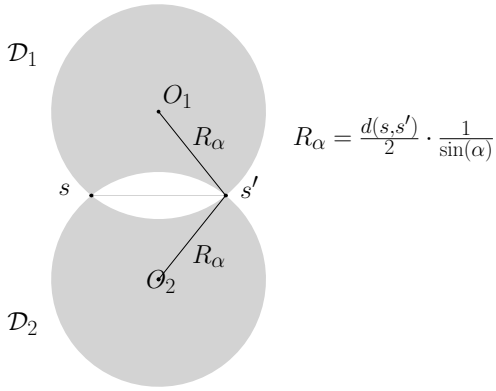


Figure 3.9: Any two distinct sensors $s \neq s'$ uniquely determine, for a given $\alpha \in (0, \pi/2]$, two disks $\mathcal{D}_1 = \mathcal{D}(O_1, R_\alpha)$ and $\mathcal{D}_2 = \mathcal{D}(O_2, R_\alpha)$ of similar radius that have s and s' on their boundary. The set of targets that are α-covered by the (s, s') is exactly $(\mathcal{D}_1 \cup \mathcal{D}_2) \setminus (\mathcal{D}_1 \cap \mathcal{D}_2)$. The radius of the disks depends on α and is $R_\alpha = \frac{d(s,s')}{2} \cdot \frac{1}{\sin \alpha}$.

Given a set system with VC dimension d, the original ϵ-net observations of Haussler and Welzl [108] and Brönnimann and Goodrich [109] can be used to construct

a set cover of size at most $\mathcal{O}(d \log(d \cdot k^*) \cdot k^*)$. When the dimension is finite, this leads to a $\mathcal{O}(\log k^*)$-approximation. In our context, this translated into a $\mathcal{O}(k_{\text{OPT}} \log k_{\text{OPT}})$-approximation for α-AngCover. We note that the same results also apply for (α, R)-AngDist and α-AngArt, since the appropriate set systems would be obtained by intersecting the symmetric difference with two disks of radius R centered at s and s' (in the case of (α, R)-AngDist) or with the visibility polygons of s and s' (in the case of α-AngArt). Since both circles and visibility polygons have finite VC dimension, we have that the corresponding set systems for (α, R)-AngDist and α-AngArt also have finite VC dimension and as such, the $\mathcal{O}(k_{\text{OPT}} \log k_{\text{OPT}})$-approximation still holds.

Geometry of the sets. We now proceed to prove our claim that the set of targets that two sensors α-cover is induced by the symmetric difference of two disks. Let those two sensors be denoted as S_1 and S_2. We are interested in determining the set of target locations T for which $\angle(S_1 T S_2) \in [\alpha, \pi - \alpha]$.

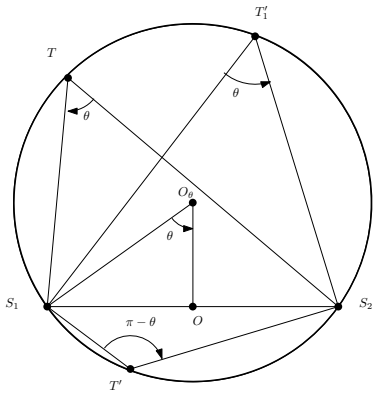


Figure 3.10: The circle centered at O_θ that passes through S_1 and S_2. All the points on the same arc (minor or major) between S_1 and S_2 will have the same angle, either θ or $\pi - \theta$.

First, we will fix a value θ and ask the question of what is the set of target locations for which $\angle(S_1 T S_2) \in \{\theta, \pi - \theta\}$. Consider an arbitrary circle centered O_θ (to be determined later) that passes through S_1 and S_2. Notice that all the points on the same arc (minor or major) between S_1 and S_2 will have the same angle with respect to S_1 and S_2, as shown in Figure 3.10. Furthemore, if the acute angle corresponding to

one of the arcs is some β, then the obtuse angle corresponding to the other arcs will be $\pi - \beta$.

The question then becomes that of determining where the center O_θ should be placed in order for the respective angle to be θ. First, notice that it is always going to be on the perpendicular bisector of S_1S_2, where O is the midpoint of the segment. We also know that S_1O_θ is going to be the radius of the circle and if $\angle S_1TS_2 = \theta$, then $\angle S_1O_\theta O = \frac{\angle S_1O_\theta S_2}{2} = \theta$ and the location of O_θ is given by

$$O_\theta O = S_1O_\theta \cdot \cos\theta.$$

Since $S_1O_\theta = \frac{S1S_2}{2\sin\theta}$, we get that

$$O_\theta O = \frac{S_1S_2}{2} \cdot \cot\theta.$$

At this point, we note that this set of equations defines two symmetric choices for O_θ, one in each of the halfspaces defined by the line S_1S_2. In general, we will denote the choice for the upper halfspace to be O_θ and the lower half as O'_θ. We denote the upper circle by

$$\mathcal{C}_\theta = \mathcal{C}(O_\theta, \tfrac{S_1S_2}{2} \cdot \tfrac{1}{\sin\theta}),$$

and similarly $\mathcal{C}'_\theta = \mathcal{C}(O'_\theta, \frac{S_1S_2}{2} \cdot \frac{1}{\sin\theta})$. In this context, we get that:

Lemma 3.12. *For any point $T \in \mathcal{C}_\theta \cup \mathcal{C}'_\theta$, the angle $\angle S_1TS_2 \in \{\theta, \pi - \theta\}$.*

Now look at a point T for which $\angle S_1TS_2 = \theta$ and consider the circumscribed circle for the triangle $\triangle TS_1S_2$. A similar analysis of its radius and center location reveals that it is exactly either $\mathcal{C}_\theta$ or $\mathcal{C}'_\theta$. Moreover, the same holds true when we require that $\angle S_1TS_2 = \pi - \theta$. In other words, we get that:

Lemma 3.13. *The set of points T for which $\angle S_1 T S_2 \in \{\theta, \pi - \theta\}$ is exactly $\mathcal{C}_\theta \cup \mathcal{C}'_\theta$.*

We can therefore focus on just describing the circles $\mathcal{C}_\theta$ for which $\theta \in [\alpha, \pi/2]$. Because of the monotonicity of the cot function, we get that:

$$0 \leq OO_\theta \leq \tfrac{S_1S_2}{2} \cdot \cot \alpha.$$

In other words, the set of all targets that can be α-covered by S_1, S_2 is exactly $\bigcup_{\theta=\alpha}^{\theta=\pi/2} \mathcal{C}_\theta \cup \mathcal{C}'_\theta$.

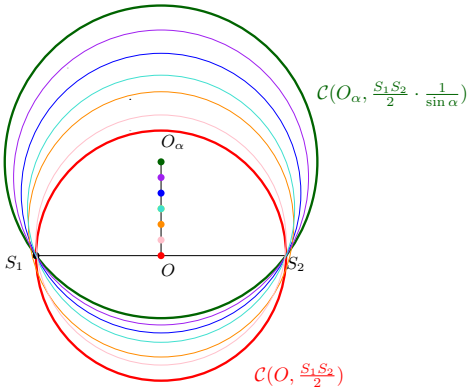


Figure 3.11: The union of circles $\mathcal{C}_\theta = \mathcal{C}(O_\theta, \frac{S_1S_2}{2} \cdot \frac{1}{\sin\theta})$ for $\theta \in [\alpha, \pi/2]$ is the symmetric difference between $\mathcal{D}_\alpha = \mathcal{D}(O_\alpha, \frac{S_1S_2}{2} \cdot \frac{1}{\sin\alpha})$ and $\mathcal{D}_{\pi/2} = \mathcal{D}(O, \frac{S_1S_2}{2})$.

We investigate this union by first looking at $\bigcup_{\theta=\alpha}^{\theta=\pi/2} \mathcal{C}_\theta$. As can be seen in Figure 3.11, it turns out that this set is exactly the symmetric difference between $\mathcal{D}_\alpha$ and $\mathcal{D}_{\pi/2}$, where $\mathcal{D}_\alpha$ and $\mathcal{D}_{\pi/2}$ are the disks associated with the circles $\mathcal{C}_\alpha$ and $\mathcal{C}_{\pi/2}$:

Lemma 3.14. *For any $\alpha \in [0, \pi/2]$, we have that: $\bigcup_{\theta=\alpha}^{\theta=\pi/2} \mathcal{C}_\theta = \mathcal{D}_\alpha \cup \mathcal{D}_{\pi/2} \setminus (\mathcal{D}_\alpha \cap \mathcal{D}_{\pi/2})$.*

Proof. We will prove this statement by considering a general point in $\mathcal{D}_\alpha \cup \mathcal{D}_{\pi/2} \setminus (\mathcal{D}_\alpha \cap \mathcal{D}_{\pi/2})$ and showing that it belongs to the boundary of some circle $C(O_\theta, \frac{S_1S_1}{2} \cdot \frac{1}{\sin\theta})$ for $\theta \leq \alpha \leq \pi/2$. The other direction of inclusion is straightforward.

Let $|S_1S_2| = d$ and w.l.o.g. consider the coordinate system that has 0 as the origin. Let S_1 have coordinates $(-\frac{d}{2}, 0)$ and S_2 have $(\frac{d}{2}, 0)$. The center O_θ will then have the coordinate $(0, \frac{d}{2}\cot\theta)$. An arbitrary point $T(x, y) \in \mathcal{D}_\alpha \cup \mathcal{D}_{\pi/2} \setminus (\mathcal{D}_\alpha \cap \mathcal{D}_{\pi/2})$ will satisfy one of the following conditions:

Case 1. It is in $\mathcal{D}_\alpha$ but not in $\mathcal{D}_{\pi/2}$, in which case we have that:

$$x^2 + \left(y - \frac{d}{2}\cot\alpha\right)^2 \leq \left(\frac{d}{2}\cdot\frac{1}{\sin\alpha}\right)^2 \tag{3.1}$$

and

$$x^2 + y^2 > \left(\frac{d}{2}\right)^2 \tag{3.2}$$

Case 2. It is in $\mathcal{D}_{\pi/2}$ but not in $\mathcal{D}_\alpha$, in which case we have that:

$$x^2 + (y - \frac{d}{2}\cot\alpha)^2 > \left(\frac{d}{2}\cdot\frac{1}{\sin\alpha}\right)^2 \tag{3.3}$$

and

$$x^2 + y^2 \leq \left(\frac{d}{2}\right)^2 \tag{3.4}$$

Given such a point $T(x, y)$, we will look at the circumscribed circle of $\triangle TS_1S_2$. Naturally, this angle will correspond to a particular $\mathcal{C}(O_\theta, \frac{S_1S_2}{2}\cdot\frac{1}{\sin\theta})$, as we have remarked before. So we really need to check that $\theta \in [\alpha, \pi/2]$. This is equivalent to checking whether the radius $R_\theta = \frac{S_1S_2}{2}\cdot\frac{1}{\sin\theta}$ is smaller than the radius $R_\alpha = \frac{d}{2}\cdot\frac{1}{\sin\alpha}$. We also know that the radius of the circumscribed circle satisfies the formula:

$$R_\alpha = \frac{S_1T\cdot S_2T\cdot S_1S_2}{4A_{\triangle S_1TS_2}},$$

where $A_{\triangle S_1TS_2}$ is the area of $\triangle S_1TS_2$. Since $A_{\triangle S_1TS_2} = \frac{1}{2}S_1S_2\cdot|y|$, we get that:

$$R_\alpha = \frac{\sqrt{(x+\frac{d}{2})^2+y^2}\cdot\sqrt{(x-\frac{d}{2})^2+y^2}}{2|y|}.$$

So in order to check that $\theta \in [\theta, \pi/2]$, we need to check that the following inequality holds:

$$\frac{\sqrt{(x+\frac{d}{2})^2+y^2}\cdot\sqrt{(x-\frac{d}{2})^2+y^2}}{2|y|} \leq \frac{d}{2}\cdot\frac{1}{\sin\alpha}. \tag{3.5}$$

Therefore, we must show that in the first case, properties (3.1) and (3.2) imply (3.5) and, in the second case, (3.3) and (3.4) also imply (3.5).

First, notice that (3.1) and (3.2) require that $y \geq 0$. This is because, if $y < 0$, we would have that:

$$\begin{aligned} x^2 + \left(y - \frac{d}{2}\cot\alpha\right)^2 &= x^2 + y^2 + \left(\frac{d}{2}\cot\alpha\right)^2 - 2y\frac{d}{2}\cot\alpha \\ &>= x^2 + y^2 + \left(\frac{d}{2}\cot\alpha\right)^2 \\ &> \left(\frac{d}{2}\right)^2 + \left(\frac{d}{2}\cot\alpha\right)^2 \\ &= \left(\frac{d}{2}\cdot\frac{1}{\sin\alpha}\right)^2, \end{aligned}$$

which is a contradiction with (3.1). A similar argument shows that (3.3) and (3.4) require that $y \leq 0$. Notice that the sign of x does not matter in our properties or the rest of the proof, so we can assume without loss of generality that $x \geq 0$.

Let us first consider the first case in which (3.1) and (3.2) are true. Because $y \geq 0$, we are left to show that:

$$\left(\left(x + \frac{d}{2}\right)^2 + y^2\right) \cdot \left(\left(x - \frac{d}{2}\right)^2 + y^2\right) \leq 4y^2 \cdot \left(\frac{d}{2}\cdot\frac{1}{\sin\alpha}\right)^2.$$

Define the function $f(y) = \left(\left(x + \frac{d}{2}\right)^2 + y^2\right) \cdot \left(\left(x - \frac{d}{2}\right)^2 + y^2\right) - 4y^2 \cdot \left(\frac{d}{2}\cdot\frac{1}{\sin\alpha}\right)^2$. Then the inequality to prove becomes $f(y) \leq 0$. We can rewrite the function the following way:

$$f(y) = y^4 + y^2 \cdot g(x) + h(x),$$

where $g(x) = 2x^2 + 2\left(\frac{d}{2}\right)^2 - 4\left(\frac{d}{2}\right)^2\frac{1}{\sin^2\alpha}$ and $h(x) = \left(x^2 - \left(\frac{d}{2}\right)^2\right)^2$. Notice that $g(x)$ is always increasing, but $h(x)$ is increasing for $x \geq \frac{d}{2}$ and decreasing otherwise.

Suppose $x \geq \frac{d}{2}$ and notice from (3.1) that:

$$\begin{aligned} x^2 &\leq \left(\frac{d}{2} \cdot \frac{1}{\sin\alpha}\right)^2 - \left(y - \frac{d}{2}\cot\alpha\right)^2 \\ &= \left(\frac{d}{2}\right)^2 - y^2 + 2y \cdot \frac{d}{2}\cot\alpha. \end{aligned}$$

We then have that $g(x) \leq g\left(\sqrt{\left(\frac{d}{2}\right)^2 - y^2 + 2y \cdot \frac{d}{2}\cot\alpha}\right) = -4\left(\frac{d}{2}\right)^2\cot^2\alpha - 2y^2 + 4y \cdot \frac{d}{2}\cot\alpha$ and $h(x) \leq h\left(\sqrt{\left(\frac{d}{2}\right)^2 - y^2 + 2y \cdot \frac{d}{2}\cot\alpha}\right) = y^4 + 4y^2 \cdot \left(\frac{d}{2}\right)^2\cot^2\alpha - 4y^3 \cdot \frac{d}{2}\cot\alpha$. Therefore, we have that:

$$\begin{aligned} f(y) &\leq y^4 + y^2 \cdot g\left(\sqrt{\left(\frac{d}{2}\right)^2 - y^2 + 2y \cdot \frac{d}{2}\cot\alpha}\right) + h\left(\sqrt{\left(\frac{d}{2}\right)^2 - y^2 + 2y \cdot \frac{d}{2}\cot\alpha}\right) \\ &= 0. \end{aligned}$$

Now suppose $x < \frac{d}{2}$. Then the function $h(x)$ is decreasing. From (3.2) we have that $x^2 \leq (\frac{d}{2})^2 - y^2$, so then we get that $h(x) \leq h\left(\sqrt{\left(\frac{d}{2}\right)^2 - y^2}\right) = y^4$ and

$$\begin{aligned} f(y) &\leq y^4 + y^2 \cdot g\left(\sqrt{\left(\frac{d}{2}\right)^2 - y^2 + 2y \cdot \frac{d}{2}\cot\alpha}\right) + h\left(\sqrt{\left(\frac{d}{2}\right)^2 - y^2}\right) \\ &= 4y^2 \cdot \frac{d}{2}\cot\alpha \cdot \left(y - \frac{d}{2}\cot\alpha\right). \end{aligned}$$

If $y \leq \frac{d}{2}\cot\alpha$, then it follows that $f(y) \leq 0$.

So now we need to consider the case when $y \geq \frac{d}{2}\cot\alpha$. For this case, we will look at $f(y)$ as an equation of degree 2 in y^2. Notice that:

$$\begin{aligned} \Delta &= g^2(x) - 4h(x) \\ &= 16\left(\frac{d}{2}\right)^2\cot^2\alpha \cdot \left[\left(\frac{d}{2}\right)^2 \frac{1}{\sin^2\alpha} - x^2\right]. \end{aligned}$$

Since $x \leq \frac{d}{2} \leq \frac{d}{2}\frac{1}{\sin\alpha}$, we get that $\Delta \geq 0$. Let $Y = y^2$ and notice that the solutions

to the equation $f(Y) = 0$ are:

$$\begin{aligned} Y_{1,2} &= \frac{-g(x) \pm \sqrt{\Delta}}{2} \\ &= -x^2 - \left(\frac{d}{2}\right)^2 + 2\left(\frac{d}{2}\right)^2 \frac{1}{\sin^2\alpha} \pm 2\frac{d}{2}\cot\alpha \cdot \sqrt{\left(\frac{d}{2}\right)^2 \frac{1}{\sin^2\alpha} - x^2}. \end{aligned}$$

In order to guarantee that $f(Y) \leq 0$, we need to show that $Y_2 \leq Y \leq Y_1$.

First, let us show that $Y \leq Y_1$. Notice that (3.1) implies that

$$Y = y^2 \leq \left(\frac{d}{2}\cot\alpha + \sqrt{\left(\frac{d}{2}\right)^2 \frac{1}{\sin^2\alpha} - x^2}\right)^2,$$

in which we inherently used that $\frac{d}{2}\cot\alpha \leq y$.

It turns out that $\left(\frac{d}{2}\cot\alpha + \sqrt{(\frac{d}{2})^2 \frac{1}{\sin^2\alpha} - x^2}\right)^2 = Y_1$ so we get that $Y \leq Y_1$.

Now we will show that $Y \geq Y_2$. Notice that (3.2) implies that

$$Y = y^2 \geq -x^2 + (\frac{d}{2})^2,$$

so we only need to show that $Y_2 \leq -x^2 + (\frac{d}{2})^2$. Notice:

$$-x^2 - \left(\frac{d}{2}\right)^2 + 2\left(\frac{d}{2}\right)^2 \frac{1}{\sin^2\alpha} - 2\frac{d}{2}\cot\alpha \cdot \sqrt{\left(\frac{d}{2}\right)^2 \frac{1}{\sin^2\alpha} - x^2} \leq -x^2 + \left(\frac{d}{2}\right)^2 \iff$$

$$2\left(\frac{d}{2}\right)^2 \left(\frac{1}{\sin^2\alpha} - 1\right) = 2\left(\frac{d}{2}\right)^2 \cot^2\alpha \leq 2 \cdot \frac{d}{2}\cot\alpha \cdot \sqrt{\left(\frac{d}{2}\right)^2 \frac{1}{\sin^2\alpha} - x^2} \iff$$

$$\frac{d}{2}\cot\alpha \leq \sqrt{\left(\frac{d}{2}\right)^2 \frac{1}{\sin^2\alpha} - x^2} \iff$$

$$x^2 \leq \left(\frac{d}{2}\right)^2 \cdot \left(\frac{1}{\sin^2\alpha} - \cot^2\alpha\right) = \left(\frac{d}{2}\right)^2,$$

which is true by our initial assumption.

Now we will consider the second case, in which (3.3) and (3.4) are true. Remember that in this case, we have that $y \leq 0$ so we denoting $y_0 = |y|$, the equations

(3.3), (3.4) become:

$$x^2 + \left(y_0 + \frac{d}{2}\cot\alpha\right)^2 > \left(\frac{d}{2}\cdot\frac{1}{\sin\alpha}\right)^2 \tag{3.6}$$

$$x^2 + y_0^2 \leq \left(\frac{d}{2}\right)^2 \tag{3.7}$$

and claim (3.5) remains as before:

$$f(y_0) = \left(\left(x + \frac{d}{2}\right)^2 + y_0^2\right)\cdot\left(\left(x - \frac{d}{2}\right)^2 + y_0^2\right) - 4y_0^2\cdot\left(\frac{d}{2}\cdot\frac{1}{\sin\alpha}\right)^2 \leq 0.$$

Notice that it follows from (3.7) that $x \leq \frac{d}{2}$, so again $\Delta \geq 0$. By a similar argument as before, we get from (3.6) that:

$$y_0 \geq -\frac{d}{2}\cot\alpha + \sqrt{\left(\frac{d}{2}\right)^2\frac{1}{\sin^2\alpha} - x^2} \geq 0$$

because $x \leq \frac{d}{2}$. So we can safely say that:

$$Y_0 = y_0^2 \geq \left(-\frac{d}{2}\cot\alpha + \sqrt{\left(\frac{d}{2}\right)^2\frac{1}{\sin^2\alpha} - x^2}\right)^2$$

$$= Y_2.$$

On the other hand, from (3.7) we get that $Y_0 \leq (\frac{d}{2})^2 - x^2$ which can be shown to be $\leq Y_1$.

□

Notice that we have initially assumed that all the centers are situated in the same half plane defined by S_1S_2. However, our proof works for the opposite half plane as well, where we consider

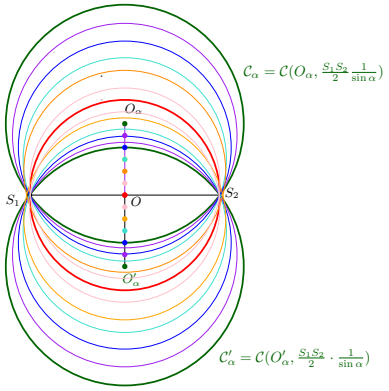


Figure 3.12: The union of circles $\mathcal{C}_\theta = \mathcal{C}(O_\alpha, \frac{S_1S_2}{2}\cdot\frac{1}{\sin\alpha})$ and $\mathcal{C}' = \mathcal{C}(O'_\alpha, \frac{S_1S_2}{2}\cdot\frac{1}{\sin\alpha})$

the symmetric center O'_α. In this case,

we get the same type of result as before:

$$\bigcup_{\alpha=\theta}^{\alpha=\pi/2} \mathcal{C}'_\theta = \mathcal{D}'_\alpha \cup \mathcal{D}_{\pi/2} \setminus (\mathcal{D}'_\alpha \cap \mathcal{D}_{\pi/2}).$$

Taking the union over all circles in both half spaces, we get that their union is the symmetric difference of $\mathcal{D}_\alpha$ and $\mathcal{D}'_\alpha$, as seen in Figure 3.12. Formally, claim that:

Lemma 3.15. *For any $\alpha \in [0, \pi/2]$, we have that:*

$$\bigcup_{\alpha=\theta}^{\alpha=\pi/2} \mathcal{C}_\theta \cup \bigcup_{\alpha=\theta}^{\alpha=\pi/2} \mathcal{C}'_\theta = \mathcal{D}_\alpha \cup \mathcal{D}'_\alpha \setminus (\mathcal{D}_\alpha \cap \mathcal{D}'_\alpha).$$

Proof. Rewriting the symmetric difference formulas from before, we get that

$$\bigcup_{\alpha=\theta}^{\alpha=\pi/2} \mathcal{C}_\theta \cup \bigcup_{\alpha=\theta}^{\alpha=\pi/2} \mathcal{C}'_\theta = (D_\alpha \setminus D_{\pi/2}) \cup (D_{\pi/2} \setminus D_\alpha) \cup (D'_\alpha \setminus D_{\pi/2}) \cup (D_{\pi/2} \setminus D'_\alpha).$$

Our claim then becomes:

$$(D_\alpha \setminus D_{\pi/2}) \cup (D_{\pi/2} \setminus D_\alpha) \cup (D'_\alpha \setminus D_{\pi/2}) \cup (D_{\pi/2} \setminus D'_\alpha) = (D_\alpha \setminus D'_\alpha) \cup (D'_\alpha \setminus D_\alpha).$$

We will prove this claim by showing that

$$(D_\alpha \setminus D_{\pi/2}) \cup (D_{\pi/2} \setminus D'_\alpha) = D_\alpha \setminus D'_\alpha.$$

The other side of the equality will follow by symmetry.

Consider a point $T(x, y)$ in $D_\alpha \setminus D_{\pi/2}$, i.e. satisfying (3.1) and (3.2). We will then show that it is contained in $D_\alpha \setminus D'_\alpha$. We have shown before this implies that

$y \geq 0$. Now we need to show that $T \notin D'_\alpha$. This is because:

$$\begin{aligned} x^2 + \left(y + \frac{d}{2}\cot\alpha\right)^2 &\geq x^2 + y^2 + \left(\frac{d}{2}\right)^2 \cot^2\alpha \\ &> \left(\frac{d}{2}\right)^2 + \left(\frac{d}{2}\right)^2 \cot^2\alpha \\ &= \left(\frac{d}{2}\right)^2 \frac{1}{\sin^2\alpha}. \end{aligned}$$

Now consider the case when $T \in D_{\pi/2} \setminus D'_\alpha$. By symmetry, we have that this implies $y \geq 0$. We need to show that $T \in D_\alpha$. From property (3.4), we have that that

$$\begin{aligned} x^2 + \left(y - \frac{d}{2}\cot\alpha\right)^2 &\leq x^2 + y^2 + \left(\frac{d}{2}\right)^2 \cot^2\alpha \\ &= \left(\frac{d}{2}\right)^2 + \left(\frac{d}{2}\right)^2 \cot^2\alpha \\ &= \left(\frac{d}{2}\right)^2 \frac{1}{\sin^2\alpha}. \end{aligned}$$

We therefore get that

$$(D_\alpha \setminus D_{\pi/2}) \cup (D_{\pi/2} \setminus D'_\alpha) \subseteq D_\alpha \setminus D'_\alpha.$$

Now consider a point $T \in D_\alpha \setminus D'_\alpha$. We have two cases:

Case 1. If $T \in D_{\pi/2}$, then because $T \notin D'_\alpha$, we have that $T \in D_{\pi/2} \setminus D'_\alpha$.

Case 2. If $T \notin D_{\pi/2}$, then because $T \in D_\alpha$, we have that $T \in D_\alpha \setminus D_{\pi/2}$.

In other words, we get

$$D_\alpha \setminus D'_\alpha \subseteq (D_\alpha \setminus D_{\pi/2}) \cup (D_{\pi/2} \setminus D'_\alpha).$$

□

3.6.2 The Pairwise Selection Formulation

In this section, we would like to switch the perspective from the previous section in the following way: instead of considering the elements of the universe to be the targets, we consider them to be the sensors. Specifically, consider the graph $G = (V, E)$ with vertex set $V = X$ and the edge set E to be such that there exists an edge $(s, s') \in E$ between two sensors $s, s' \in V$ if they α-cover a target.

For each target $t \in T$, we look at all the pairs of sensors that α-cover it. By definition, this will be a subset E_t of the edges in E, i.e. a subset of the pairs of sensors. The task of picking the smallest number of sensors that α-cover T then becomes equivalent to picking the smallest set of vertices $S \subseteq V$ with the property that, for each target $t \in T$, there exist two sensors $s, s' \in S$ such that $(s, s') \in E_t$. We formalize this as the PAIRWISE SELECTION problem:

Pairwise Selection

Input : A graph $G = (V, E)$ and a collection $E_1, E_2, \ldots, E_l$ of l subsets of edges, $E_i \subseteq E$, for all $i \in \overline{1, l}$.

Output : A set $S \subseteq V$ of mininum cardinality such that $E[S] \cap E_i \neq \emptyset$, for all $i \in \overline{1, l}$.

By $E[S]$ we mean the edges induced by the subset S, i.e. an edge $e = (u, v)$ is in $E[S]$ if both of its endpoints are in S, $u, v \in S$. In this context, we say that the edge (u, v) is *satisfied* or, more specifically, the edge (u, v) satisfies the set E_i if u and v are both chosen in the set and $(u, v) \in E_i$.[1]

[1]This is because we can associate with each vertex v a boolean variable x_v and then each set

We now show that the MIN REP problem is a special case of PAIRWISE SELECTION. In MIN REP, we are given a bipartite graph $G = (A, B, E)$, where $|A| = |B| = n$. Each of the sets A and B are partitioned in k sets of size $q = \frac{n}{k}$ each, $\mathcal{A} = \{A_i | i \in \{1, \ldots, k\}\}$ and $\mathcal{B} = \{B_i | i \in \{1, \ldots, k\}\}$. Given this, we form a the bipartite *supergraph* H in which the vertices represent the sets A_i and B_i. Vertices A_i and B_j are connected by a *superedge* if there exist elements $a_i \in A_i$ and $b_j \in B_j$ that are adjacent in G. In this situation, we say that the edge (a_i, b_j) *covers* the superedge (A_i, B_j). The MIN REP problem asks for the set $S = A' \cup B'$ of minimum size such that the pairs $(a', b'), a' \in A', b' \in B'$ cover all the superedges of H.

Any such instance of MIN REP can be transformed into an instance of PAIRWISE SELECTION in the following way: for each superedge (A_i, B_j) define the set

$$E_{i,j} = \{(a, b) | a \in A_i, b \in B_j\}.$$

In this construction, a min size solution to MIN REP is equivalent to a minimum size solution to PAIRWISE SELECTION. In other words, if there exists an α-approximation algorithm for PAIRWISE SELECTION, then we would immediately get an α-approximation algorithm for MIN REP. Kortsarz [122] showed, however, that there is a hardness of $2^{\log^{1-\epsilon} n}$, for any $0 < \epsilon < 1$ for MIN REP unless NP $\subseteq$ DTIME($n^{\text{polylog}(n)}$), and hence, for PAIRWISE SELECTION as well. In terms of positive results, there is a $O(n^{1/3} \log^{2/3} n)$-approximation algorithm for MIN REP due to Charikar et al [134]. We believe it would be an interesting challenge to try to extend it to PAIRWISE SELECTION.

E_i can be described as the clause $\bigvee_{(u,v)=e \in E_i} (x_v \wedge x_u)$.

3.6.3 Dominating Set Reduction

Given the Pairwise Selection problem we defined in the previous section, it is clear how instances of α-Ang etc. can be represented as special cases of it. The hardness of approximation results, however, do not follow. In fact, as we have shown, these problems admit $\mathcal{O}(k_{\text{OPT}} \log k_{\text{OPT}})$- approximations as instances of Set Cover.

Such approximations were obtained as a consequence of exploiting the inherent geometry of the problems. In general, it is believed that geometric instances are normally easier to solve than their more general counterparts. This essentially comes from the fact that, in general, it is hard or even impossible to construct Euclidean instances that are equivalent to general instances (which might not even satisfy the triangle inequality). In fact, in the case of Pairwise Selection, it is unclear even how to incorporate the notion of geometry in the problem formulation, i.e. the sets $E_1, \ldots, E_l$ can be determined arbitrarily.

Nevertheless, one way in which we can encode the geometry of the problem is by considering *planar* graphs. Roughly speaking, a planar graph is a graph that can be drawn on the sphere without any of its edges crossing. One could use a planar graph as an instance of a geometric graph and use whatever complexity results are available for planar graphs to reduce from. In particular, Garey et al [92] have shown that the Dominating Set problem is NP-complete on planar graphs even when restricted to planar graphs of degree at most 3. In the Dominating Set problem, we are looking for a dominating set of minimum size. Furthermore, Valiant [135]

has given a transformation that embeds any planar graph of degree at most 4 in the plane in such a way that its vertices have integer coordinates and its edges are vertical and horizontal line segments.

At this point, we can trace what a possible reduction for α-Ang etc. could look like: given a general instance of the Dominating Set problem on planar graphs of degree at most 3, apply the transformation of Valiant [135] and obtain an embedding of the graph in the plane. The next step would be to then construct an instance of α-Ang etc. that is equivalent to the given Dominating Set instance. We would then obtain that α-Ang etc are NP-complete. To that end, we present an intermediary ingredient in the reduction: given an instance of Dominating Set, we present an equivalent instance of Pairwise Selection. In particular, this reduction can be used to show that:

Theorem 3.9. Pairwise Selection *is NP-complete on planar graphs of degree at most* 3.

We note that the reduction does not exploit the geometry of the input graph and as such, we believe it can be used as a blueprint for proving NP-completeness or α-Ang etc in the following way: consider an instance of Dominating Set on planar graphs of degree ≤ 3, apply the transformation of Valiant [135] and construct an instance of α-Ang etc that satisfies geometrically the instance of Pairwise Selection we construct. Specifically, we need to make sure that the sets E_t which we construct for each target can actually be realized by the constraints in α-Ang etc. For example, if, in the construction, we require a set $E_t = \{(s_1, s_2)\}$,

we need to make sure that, in the transformed graph, the target t is α-covered only by the pair of sensors (s_1, s_2) and by no other pair of sensors in X.

We now proceed with the actual reduction.

Construction:

Given an instance $G = (V, E)$ of the DOMINATING SET problem, we construct the following instance of the PAIRWISE SELECTION problem. For each vertex $u \in V$, we add an auxiliary vertex u'. Let the set of such auxiliary vertices be called V'. Throughout the remainder of the proof, we will denote vertices in V exclusively by single letters, i.e. u, and their associated auxiliary vertices as the corresponding letter followed by an apostrophe, i.e. u' represents the auxiliary vertex of u.

For each vertex $u \in V$, we define an edge between u and its corresponding auxiliary vertex $u' \in V'$. In other words, we define the set $E' = \{(u, u') | u \in V\}$. The input graph to the PAIRWISE SELECTION problem will be $G = (V \cup V', E \cup E')$. We will refer to the original edges in E as being *red* edges and the newly formed edges in E' as being *blue*. For each vertex $u \in V$, we denote the neighborhood of u in G by $N_G(u) = \{v | \exists e \in E, e = (u, v)\}$. For each vertex $u \in V$, we define the set

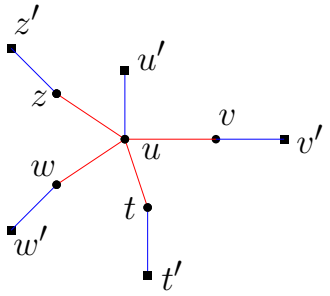


Figure 3.13: Suppose v, w, t, z are neighbors of u in G. Then E_u will consist of all the red edges that u particsipates in, plus all the blue edges that are associated with u and its neighbors.

$$E_u = \{e \in E | u \in e\} \cup \{(u, u')\} \cup \{(v, v') | v \in N_G(u)\}.$$

Dominating set of size k $\rightarrow$ pairwise selection of size $2k$:

Proof. Let $D \in V$ be a dominating set in G. Define $S = D \cup D'$ where $D' = \{u' | u \in D\}$. We claim that S is a valid pairwise selection for the graph G'. Consider an arbitrary vertex $u \in V$ and its associated set E_u. We distinguish two cases:

Case 1. u was selected in the dominating set. Then both u and u' are in S, and therefore, the edge $(u, u') \in E[S]$. It follows that $(u, u') \in E[S] \cap E_u \neq \emptyset$.

Case 2. u was not selected in the dominating set. But then there must exist a vertex $v \in N_G(u)$ such that $v \in D$. Then $v, v' \in S$ and $(v, v') \in E[S] \cap E_u \neq \emptyset$.

Therefore, S forms a valid pairwise selection and since $|D| = k$, we have that $|S| = 2k$. □

Pairwise selection of size $k \rightarrow$ dominating set of size $\lfloor k \rfloor$:

Proof. Let S be a pairwise selection of size k. First, notice that if $u' \in S$, then it must be the case that either $u \in S$ or there exists a smaller pairwise selection of size $k - 1$. If $u \notin S$, then u' cannot contribute to any edge in $E[S]$ and therefore can be removed from the set. Right now, it is not clear whether the same holds for vertices $u \in S$: it could be the case that $u' \notin S$. We will argue, however, that any pairwise selection S in which this happens can be transformed into another pairwise selection S^*, of smaller or equal size, in which, for every $u \in S^*$, we have that $u' \in S^*$. In other words, S^* only consists of vertices u and their auxiliary vertices u'. We call a pairwise selection with this property **symmetrically built**.

Before we describe that transformation, we will first show how to obtain a dominating set from a symmetrically built pairwise selection S^*. Let $D = S^* \cap V$.

We claim that this set D forms a dominating set in G. Notice that we only need concern ourselves with vertices $v \in V$ that are not in D. Look at the associated set E_v. Since S^* is a valid pairwise selection, we must have that there exists an edge in $E_v \cap E[S^*]$ that satisfies E_v. But since $E[S^*] \cap E_v$ consists only of blue edges (because all potential red edges require that $v \in S^*$)and since, $v, v' \notin S^*$, we have that there must exist another blue edge in E_v that is satisfied. But every such edge in E_v is defined by a neighbor u of v and its auxiliary vertex u'. So there must exist a neighbor $u \in S^*$. But that implies that there exists a neighbor $u \in D$. So v is adjacent to a vertex in the dominating set. Since the $|S^*| = 2|D|$, we get the appropriate equivalence.

Now we will describe how to transform any pairwise selection S into a symmetrically built pairwise selection S^* of smaller or equal size. By our initial observation, we can assume that there are no $u' \in S$ such that $u \notin S$. Let $v \in S$ be such that $v' \in S$. We call such a vertex in S a **symmetric** vertex. Every other vertex $u \in S$ that is not symmetric will be called **assymetric**. Therefore, a symmetrically built pairwise selection therefore consists only of symmetric vertices.

Intuition:

Suppose a pairwise selection S is not symmetrically built. Then it must be the case that there exists a vertex $u \in S$ that is asymmetric. The general intuition is that we will delete u from S, therefore gaining a budget of 1, and instead "activate" the auxiliary vertex of another symmetric vertex, therefore maintaining the solution size. Notice that activating auxiliary vertices does not damage our solution: it can only improve satisfiability. On the other hand, deleting a vertex might make our

solution incorrect.

At this point, it is useful to point out that deleting an asymmetric vertex u could potentially affect the satisfiability status of E_u and all E_v for $v \in N_G(u)$. This is because the edges that u is adjacent belong only to the aforementioned sets. Moreover, notice that if a neighbor $v \in N_G(u)$ has not been chosen in the pairwise selection, then removing u will not affect the satisfiability of E_v. This is because the only edges in E_v that are adjacent to u are (u, v) and (u, u'). Since $v \notin S$ and $u' \notin S$, we already know that those edges did not contribute to satisfying E_v in the first place. So deleting u will not affect them negatively.

So we will only concern ourself with neighbors of u that have been chosen in S. Notice that in order for the set E_u to be satisfied, we need at least one neighbor v of u to be in S, i.e. $\exists v \in N_G(u), v \in S$. We distinguish the following cases:

Case 1. Every neighbor of u that is included in S is symmetric. In that case, notice that u cannot contribute in satisfying any sets E_v, where $v \in V - \{u\}$. This is because, the only way u can contribute to a set E_v is if $v \in N_G(u)$ and $v \in S$. But if $v \in S$, it must be symmetric and the edge (v, v') already satisfies the set E_v. In this context, we can safely discard u from the set S and recurse on the remaining set which has 1 less asymmetric vertex and is of smaller size.

Case 2. There are some neighbors of u that are in S but are asymmetric. In this context, we would like either to add the auxiliary vertex u' to S (and therefore make u symmetric) or discard u entirely. Notice however, that regardless of which one of these operations we do, this will not affect the satisfiability status of the sets E_v where $v \in N_G(u)$ and v is symmetric (by the argument from above). The only

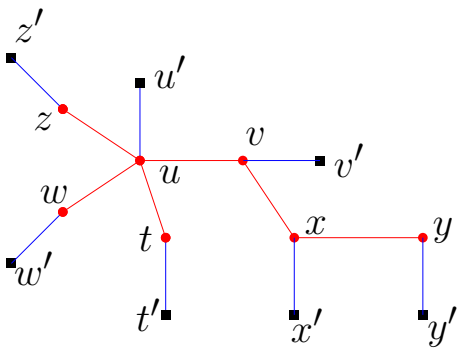


Figure 3.14: The red vertices belong to S.

potential damage could be done to sets E_v where $v \in N_G(u)$ and v is asymmetric. We can therefore restrict our attention only to sets E_v in which $v \in S$ is asymmetric.

At this point, we can try to investigate the local neighborhood of u. We are guaranteed that u has at least one neighbor v that is also asymmetric. At this point, we cannot simply afford to add u' to S because that would increase our solution size. So can we delete u instead? Suppose we do that. Then the sets affected by this are E_u, E_v and every other set E_z in which $z \in N_G(u)$ and z is asymmetric. We could activate v' and that takes care of the satisfiability of E_u and E_v. Unfortunately, that does not guarantee that E_z is satisfied as well (notice that earlier, E_z could have potentially been satisfied by the edge (u, z) which now is not satisfied any more). In order to control such a scenario, we need to investigate the global structure of asymmetric vertices. For example, in Fig. 3.14, the red vertices belong to S.

In this context, consider the subgraph of G induced by these asymmetric vertices. We shall refer to this subgraph as H. In other words, H consists of all asymmetric vertices together with the red edges that they define. Notice, however, that in the case we are considering, none of these vertices has degree 0 in H.

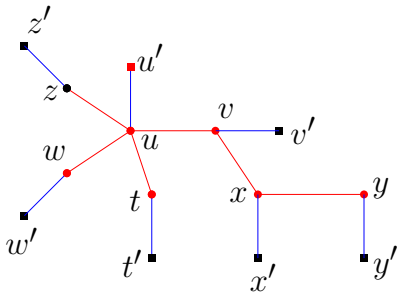


Figure 3.15: Swap z with u' in our construction.

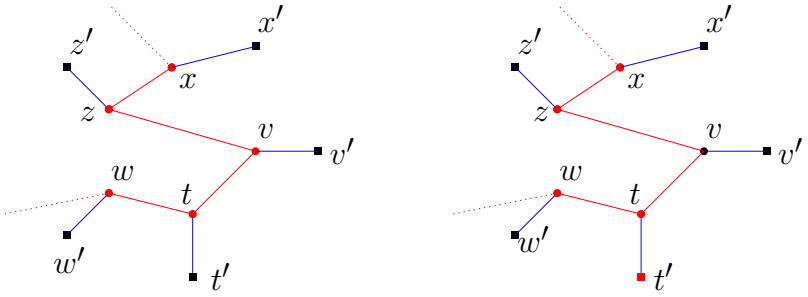
This is because we are considering asymmetric vertices whose neighbors are also asymmetric.

At this point, suppose the graph H has a vertex of degree 1. For demonstrative purpose, suppose we consider vertex z and let u be its only neighbor in H. Notice then that we can remove vertex z from S and instead replace it with u', like in Fig. 3.15.

Then the set E_z is satisfied by the edge (u, u') and u becomes symmetric. The solution size stays the same. At this point, we have reduced the set of asymmetric vertices by 2 and we can recurse.

Suppose, on the other hand, that the graph H does not have any vertices of degree 1, like in Fig. 3.16(a).

In this context, pick any vertex, say vertex v with neighbors z and t as in the figure. Then we claim that we can safely delete v from S. This is because the only sets potentially satisfied by an edge of v are E_v and the sets associated with $N_H(v)$. However, every vertex $y \in N_H(v)$ has degree at least 2, and therefore, the set E_y is already satisfied by an edge other than (y, v). In Fig. 3.16(b), the sets E_z and E_t are satisfied by the edges (z, x) and (t, w) which must exist because the degrees of z and



(a) The graph H does not have vertices of degree 1.

(b) We can bring down the size of our instance.

Figure 3.16: If the graph does not have vertices of degree 1, then we can reduce the size of our instance.

t are ≥ 2. The only other set that is affected by the disappearance of v is E_v itself. This can be solved by making one of v's neighbors symmetric. In our case, the edge (t, t') satisfies the set E_v. We therefore get a new pairwise selection instance of the same size in which we have reduced the number of asymmetric vertices by 2, like in Fig 3.16(b).

□

3.7 The Continuous Case

In this chapter, we discuss the case in which we can place the sensors anywhere on the plane. As mentioned in the introduction, the problem with trying to adapt k-CENTER methods to our problem is that, once we select sensors that can optimally cover the far apart targets, we cannot guarantee that those sensors can help cover the rest of the targets. Geing able to place sensors wherever we want, however, allows us to select an additional small set of sensors that, together with the optimal sensors, can cover a target wherever it might be.

In this context, we present a constant factor approximation for the sensor placement problem considered by Tekdas et al [120], in which they consider an upper bound U on the uncertainty function given as:

$$U(s_1, s_2, t) = \frac{d(t,s_1) \cdot d(t,s_2)}{|\sin\theta|},$$

where $\theta = \angle s_1 t s_2$. The task becomes that of selecting the smallest number of sensors that cover each target such that for each target t, there exists a pair of sensors s_1, s_2 such that $U(s_1, s_2, t) \leq U$. By tiling the plane with triangles and placing sensors at vertices of the triangular grid, Tekdas et al [120] give a 3-approximation that guarantees that the maximum uncertainty is $\leq 5.5U$. In order to avoid the bi-criteria nature of the result, we show that an alternative tiling with squares gives a 25-approximation that guarantees that we never exceed the uncertainty threshold.

The approach in both cases is similar to the k-CENTER approach described in the introduction, but at first sight, it might be unclear how the problems are similar in nature. In particular, the product $d(t, s_1) \cdot d(t, s_2)$ can be allowed to be high as long as the value for $\sin\theta$ is high as well(i.e. θ is close to $\pi/2$). Moreover, by considering the product of distances instead of the individual distances themselves, we are allowing one sensor to be really close to the target while the other one can be really far away. In terms of localization, this makes intuitive sense: a sensor that is really close to the target will produce measurements that already have high accuracy. On the other hand, k-CENTER requires all sensors to be within range of the target.

To this end, the following observation becomes essential: if $d(s_1, t) > \sqrt{U}$ and

$d(s_2, t) > \sqrt{U}$, then $U(s_1, s_2, t) > U$ for any value of θ. In other words, both sensors cannot be far away from the target at the same time. A solution to this problem will guarantee that, for each target, at least one of the sensors assigned to it will be within a distance of $\sqrt{U}$ of it. We refer to such a sensor as the *primary* sensor for t. Notice that, if two targets t_1 and t_2 are far apart in the sense that $d(t_1, t_2) > 2\sqrt{U}$, then they cannot have the same primary sensor. In other words, if we consider a maximal set $P \subseteq T$ of targets with pairwise distance $> 2\sqrt{U}$, the size of that set is a lower bound on k_{OPT}.

Our strategy will be to place 25 sensors around each target $t \in P$ and use the previous lower bound to guarantee that we never use more than $25k_{\text{OPT}}$ number of sensors in total. The necessity of placing more than one sensor per target in P comes from the fact that all the other targets (that are not in P) must also be covered. From the definition of P as a maximal set, we know that each of these targets will be within distance $\leq 2\sqrt{U}$ of a target t in P. Our strategy will therefore be to place sensors around t that will cover not just t, but any target within distance $\leq 2\sqrt{U}$ of t. In other words, we consider disks of radius $2\sqrt{U}$ centered at each target $t \in P$. Let such a disks be denoted as $\mathcal{D}_t$. Given a method that can place c sensors and cover all the targets in one $\mathcal{D}_t$, we get that the total number of sensors we use is :

$$c \times |\{D_t : t \in P\}| = c \times |P| \leq c \times k_{\text{OPT}}.$$

The last ingredient, therefore, is a method that, given a disk of radius $2\sqrt{U}$ centered at a target $t \in P$, places a constant number of sensors that can cover any target contained in that disk. To that end, Tekdas et al [120] propose a method

that uses 3 sensors s_1, s_2, s_2 and guarantees that, given a target $t' \in \mathcal{D}_t$, there exist two distinct sensors $s_i \neq s_j$ with $i, j \in \{1, 2, 3\}$ such that:

$$U(s_i, s_j, t') \leq 5.5U.$$

Their strategy is to place s_1, s_2, s_3 equally spaced on the boundary of the disk of radius $\sqrt[3]{\frac{1}{4}}\sqrt{C}$ centered at t and then show that for any t', there exists a pair that achieves an uncertainty of at most $5.5U$. Our strategy will be to place sensors that cover all targets within uncertainty exactly U. We do this by tiling the disk with squares and placing sensors at each of the vertices of the square, as seen in Figure 3.17.

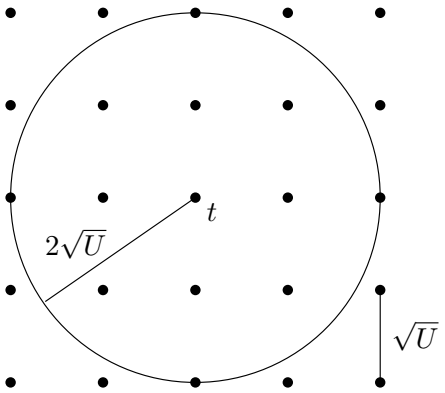


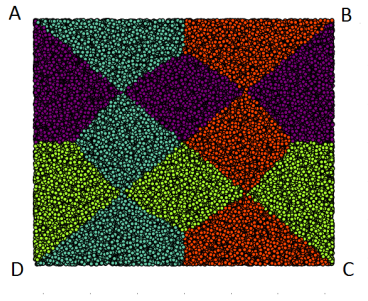
Figure 3.17: We tile the disk of radius $2\sqrt{U}$ centered at t with 16 squares of size $\sqrt{U}$. We then place sensors at each of the vertices of the squares, 25 in total.

As seen, we require $\left(\frac{4\sqrt{U}}{\sqrt{U}}\right)^2 = 16$ squares, which in total use 25 sensors. We are now left to show that, given any target in $\mathcal{D}_t$, there exists a pair of sensors that cover it within uncertainty U. We do this by showing that each square $ABCD$ can cover all the targets in its interior, as shown in Figure 3.18(a).

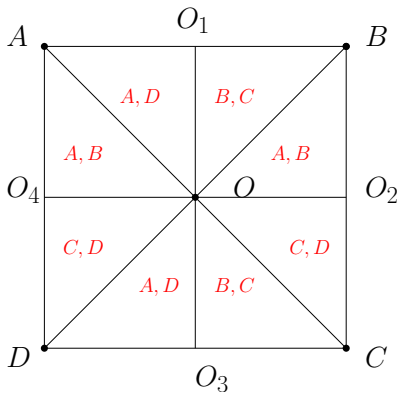
Formally, we show the following:

Lemma 3.16. *Any square $ABCD$ of side $\sqrt{U}$ that has sensors placed at its vertices can cover any target T in its interior with uncertainty $\leq U$.*

Proof. Let O be the center of $ABCD$ and O_1, O_2, O_3 and O_4 be the center's projections on each of the sides of the square. For each of the triangles described by the



(a) Given a square $ABCD$, we color code some of the pairs, in this way: (A, B)- purple, (B, C) - red , (C, D) - green, (A, D) - blue. The color map shows which of the pairs minimizes the uncertainty in which areas of the square.



(b) We split the square into 8 areas and for which of them, we assign a pair of sensors. We will claim that, for any target contained in one of the areas, the assigned pair will cover it within uncertainty U.

Figure 3.18: Partitioning of the square based on angular coverage.

points above, we assign a pair of sensors, as shown in Figure 3.18(b). For example, for the triangle $\triangle AOO_1$, we assign the pair A, D.

We claim that any potential target contained inside such a triangle will have uncertainty smaller than U when paired with the designated pair. The only exception to the rule is the vertices of the triangle themselves. If a target is located at any such location, we pair it with its 2 adjacent vertices. In that case, the uncertainty would be $\frac{\sqrt{U}\cdot\sqrt{U}}{\sin(90°)} = U$.

Let us now proceed by showing that the claim is true for the triangle $\triangle AOO_1$ and an arbitrary target T contained inside it. Let T_1, T_3 and T_4 be its projections on the AB, AD and DC sides of the square. We refer the reader to Figure 3.19 for a pictorial view of our notation.

For ease of computation, we will refer to the side length of the square as R and the length of the $\overline{AT_1}$ segment as $x \cdot R$ and the length of the $\overline{AT_4}$ as $y \cdot R$.

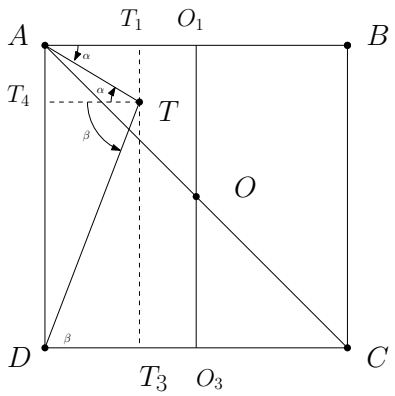


Figure 3.19: We consider a target T contained inside the triangle $\triangle AOO_1$ and its projections T_1, T_3 and T_4 on AB, AD and DC respectively. The pair assigned to this triangle is (A, D) so our task will be to show that $U(A, D, T) \leq U$.

Notice that $x \in (0, \frac{1}{2}]$ and $y \in [0, \frac{1}{2}]$ and furthermore, $y \leq x$. These two parameters define all the points in the triangle $\triangle AOO_1$, including locations on the edge AB but excluding point A. We shall also denote the angles $\angle TAT_1 = \alpha$ and $\angle TDT_3 = \beta$. Notice then that $\angle ATT_4 = \alpha$ and $\angle DTT_4 = \beta$.

Let us compute the uncertainty in terms of these parameters:

$$U(A, D, T) = \frac{AT \cdot DT}{\sin(\angle ATD)}.$$

Notice that:

$$\begin{aligned}
\sin(\angle ATD) &= \sin(\angle ATT_4 + \angle DTT_4) \\
&= \sin(\alpha + \beta) \\
&= \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta)) \\
&= \frac{TT_1}{AT} \cdot \frac{DT_3}{DT} + \frac{AT_1}{AT} \cdot \frac{TT_3}{DT} \\
&= \frac{y \cdot R}{AT} \cdot \frac{x \cdot R}{DT} + \frac{x \cdot R}{AT} \cdot \frac{(1-y) \cdot R}{DT} \\
&= \frac{xy \cdot R^2 + x(1-y) \cdot R^2}{AT \cdot DT} \\
&= \frac{x \cdot R^2}{AT \cdot DT}.
\end{aligned}$$

We get that:

$$U(A, D, T) = \tfrac{AT^2 \cdot DT^2}{x \cdot R^2}.$$

But, by Pythagora's theorem applied in the triangle $\triangle ATT_1$ and we get that:

$$AT^2 = AT_1^2 + TT_1^2 = x^2 \cdot R^2 + y^2 \cdot R^2 = (x^2 + y^2) \cdot R^2$$

Similarly, by looking at the triangle $\triangle DTT_3$, we get:

$$DT^2 = DT_3^2 + TT_3^2 = x^2 \cdot R^2 + (1-y)^2 \cdot R^2 = (x^2 + (1-y)^2) \cdot R^2$$

Overall, we have that:

$$U(A, D, T) = R^2 \cdot \tfrac{(x^2+y^2) \cdot (x^2+(1-y)^2)}{x}.$$

Remember that $R^2 = U$, so, in order to show that $U(A, D, T) \leq U$, it suffices to show that:

$$(x^2 + y^2) \cdot (x^2 + (1-y)^2) \leq x,$$

since $x > 0$. Let

$$\begin{aligned} f(x) &= (x^2 + y^2) \cdot (x^2 + (1-y)^2) - x \\ &= x^4 + x^2 \cdot (y^2 + (1-y)^2) - x + y^2 \cdot (1-y)^2 \\ &= x^4 + x^2 \cdot g(y) - x + h(y), \end{aligned}$$

where $g(y) = y^2 + (1-y)^2$ and $h(y) = y^2 \cdot (1-y)^2$. Notice that the function $g(y) = 2y^2 - 2 \cdot y + 1$ has derivative $g'(y) = 4 \cdot y - 2$ and hence is decreasing when $y \in [0, \frac{1}{2}]$. Therefore, $g(y) \leq g(0) = 1$ for $y \in [0, \frac{1}{2}]$. On the other hand, the function $h(y) = y^2 \cdot (1-y)^2 = y^4 - 2 \cdot y^3 + y^2$ has derivative

$$\begin{aligned} h'(y) &= 4 \cdot y^3 - 6 \cdot y^2 + 2 \cdot y \\ &= 2y \cdot (2y^2 - 3 \cdot y + 1) \\ &= 2y \cdot (2y - 1) \cdot (y - 1) \\ &\geq 0, \end{aligned}$$

for $y \in [0, \frac{1}{2}]$. Hence, h is increasing on the interval $[0, \frac{1}{2}]$ and since $y \leq x$, we get that $h(y) \leq h(x)$. Therefore:

$$\begin{aligned} f(x) &\leq x^4 + x^2 - x + x^2 \cdot (1-x)^2 \\ &\leq 2 \cdot x^4 - 2 \cdot x^3 + 2 \cdot x^2 - x \\ &\leq x \cdot (2 \cdot x^3 - 2 \cdot x^2 + 2 \cdot x - 1) \\ &\leq 0, \end{aligned}$$

for $x \in (0, \frac{1}{2}]$ since the function $x \cdot (2 \cdot x^3 - 2 \cdot x^2 + 2 \cdot x - 1)$ goes below 0 in between the roots 0 and 0.64780.

The same holds for points in all the other triangles and their corresponding assigned sensor pairs. The argument is similar to the one described above and exploits the appropriate symmetries. □

We have therefore shown that we can place 25 sensors and cover any target contained in $\mathcal{D}_t$ within uncertainty $\leq U$. Notice, however, that we did not make any assumptions about the positions of the targets. In fact, our result holds for the general case in which not just the sensors, but also the targets, can be located anywhere on the plane. The same situation applies as before. In other words, we have shown the following:

Theorem 3.10. *For the case when the sensors can be placed anywhere on the plane, there exists a polynomial time algorithm that uses at most* $25 \cdot k_{\mathrm{OPT}}$ *sensors and covers every target within uncertainty* $\leq U$ *(both for the case in which the target set is discrete and for when it is continous).*

The above argument could also be made for (α, R)-AngDist, in the sense that we could argue that sensors placed at A, B, C, D will collaborate to cover the targets inside the square. For that, we will be able to use the geometric representation from Section 3.6.1 and show that the union of all those symmetric differences (one for which pair of sensors) will cover the entire square. Is is unclear, however, what the side of the square should be chosen to be, i.e. it will most likely depend on α.

Chapter 4

The Traveling Salesman Problem with Neighborhoods

In this chapter, we present out contribution to the Traveling Salesman Problem with Neighborhoods on uniform disks. We begin in Section 4.1 by reviewing the related work on the TSPN in more detail, presenting our results and giving an intuitive view of our approach. We then define our terms in Section 4.2 and proceed to define the main ingredients of our result in Section 4.3. Specifically, we describe our lower bounds and structural theorems there. We then perform the analysis of the main algorithm in Section 4.4 and also discuss alternate approaches for some special cases. In Section 4.5, we present a technique for proving the Häme, Hyytiä and Hakula conjecture which helps us prove it for $n = 3$ and which we believe shows promise for the case of $n > 3$.

4.1 Introduction

In this section, we review previous work on TSPN in general, present our results and provide a high level description of our methods.

4.1.1 Related Work

The Traveling Salesman Problem with Neighborhoods (TSPN) is a generalization of the classical Euclidean Traveling Salesman Problem (TSP), when each point to be visited is replaced with a region (interchangeably, a neighborhood) and the objective is to compute a tour of minimum length that visits at least one point from each of these regions. While it is known that Euclidean TSP admits a Polynomial Time Approximation Scheme (PTAS) due to the celebrated results of Arora [136] and Mitchell [137], Euclidean TSPN has been shown in fact to be APX-hard [71,138] even for line segments of comparable length [72]. The geometric version of TSPN was first studied by Arkin and Hassin [70] who gave constant factor approximations for a variety of cases. Since then, there has been a wide ranging study of TSPN for different types of regions. In the case of connected regions, there is a series of $O(\log n)$ approximations [72, 139, 140]. Better approximations are known for cases that consider various restrictions on the regions such as comparable sizes (i.e. diameter), fatness (ratio between the smallest circumscribing radius and largest inscribing radius, or how well can a disk approximate the region) and pairwise disjointness or limited intersection [68, 71, 72, 141–146]. We refer the reader to [147]

for a comprehensive survey of the results.

Dumitrescu and Mitchell [68] were the first to specifically address the case of uniform disks in 2001. They showed a PTAS for disjoint unit disks and simpler constant factor approximations for the disjoint and overlapping cases. The specific factor of 3.547 for disjoint disks is relative to using a routine for TSP on points (i.e. the actual constant depends on the subroutine used). The PTAS and the 3.547-approximation are the best known factors for the disjoint case. Later, Dumitrescu and Tóth [143] improved the constant factor in [68] for the overlapping case from 7.62 to 6.75 and extended it to unit balls in $\mathbb{R}^d$, giving a $O(7.73^d)$-approximation. When the balls are disjoint, Elbassioni et al. [72] showed a $O(2^d/\sqrt{d})$-approximation. Most recently, Dumitrescu and Tóth [69] gave a randomized constant factor for (potentially overlapping) disks of arbitrary radii (the actual constant is not mentioned and seems large). As noted by the authors in [143], while the complexity of the disk case is well understood generally, the question of obtaining practical and better constant factor approximations remains of high interest.

4.1.2 Technical Contributions and High Level Ideas

We make the first progress on the Häme, Hyytiä and Hakula conjecture and develop a twofold method that either improves the $2Rn$ bound in (1.1) or shows that the TSP on the centers is a good approximation for the TSPN on the disks. Formally, we get that

Theorem 4.1. *For any $n \geq 4$ disjoint disks of radius R at least one of the following*

is true:

- $TSPN^*$ *is supported by a straight line,*
- $|TSP^*| \leq |TSPN^*| + 1.999Rn$*,*
- $|TSP^*| \leq 2 \cdot |TSPN^*|$*.*

Our framework also gives an overall 3.53*-approximation for the case of uniform disjoint disks and a* 6.728*-approximation for the overlapping case.*

While the improvement in the overall approximation factor is small, our framework strives to provide new insight into the problem that can be explored further. Specifically, the 2-approximation (optimal with respect to the method of computing a TSP on the centers [68]) comes from the case in which the TSPN tour takes a lot of sharp turns. Furthermore, it is based on a lower bound that does not rely on packing arguments. To the best of our knowledge, this is the first such bound specifically for TSPN out of all arguments for general fat regions [141]. As such, it might be of independent interest and it could, for example, lead to improved approximation factors for balls in $\mathbb{R}^d$ that do not depend exponentially on the dimension. Moreover, the fatness of the disks is used in showing that short sharp turns lead to a disk being visited multiple times and can conceivably be used to show similar properties for other fat regions.

We start by fixing an order σ and comparing the TSPN tour that visits the disks in that order to the TSP tour that visits their respective centers in the same order. The $2Rn$ term in (1.1) comes from considering the points at which the TSPN

touches the boundary of each disk and charging each such vertex with a $2R$ detour for going to its respective center and coming back. In this view, the $2Rn$ term cannot be improved since the charge on each vertex will always be $2R$. Instead, we reinterpret the bound as charging the *edges* of the TSPN tour instead of its vertices and notice that the charge for each edge can now be anywhere between $-2R$ and $2R$, depending on how close the tour is to (locally) visiting pairs of disks optimally. In this context, we define a "bad" edge to be one that incurs a large charge (i.e. $> (2-\epsilon)R$ for some $\epsilon > 0$). We show that such bad edges lead to the TSPN tour exhibiting sharp turns (i.e. with small interior angle). When the edges of the sharp turn are long, we use that to derive a better lower bound on the overall TSPN tour length. On the other hand, when one of them is short, we show that the tour must then visit a disk twice (i.e. visit it once, then touch another disk and return back to it). The crux of the argument is in understanding how these short sharp turns that visit a disk multiple times influence the global detour term.

When a tour visits a disk more than once, two scenarios follow naturally from the classical TSP case of just visiting points: either the order σ is not optimal or the tour must follow a straight line. Surprisingly, we show that a third alternative scenario is also possible, whose local structure we call a β-triad. The main technical contribution of the paper is in describing structural properties of such β-triads and showing that they actually have a low *average* detour. Specifically, we construct an additional order σ' and use an averaging argument to show that β-triads have low detour when compared to the TSP tours that visit the centers in the order σ and σ'. This then allows us to conclude that they have a low detour with respect to the

optimal TSP on the centers.

Along the way, we also show that the Häme, Hyytiä and Hakula conjecture is true for $n = 3$ and use it to bound the average detour of β-triads. We include a discussion of the method used to derive it, involving Fermat-Weber points, which might be useful for the case of $n \geq 4$. We also discuss how our approach can be used within the framework of Dumitrescu and Tóth [143] to yield improved approximation factors for the overlapping disks case.

4.2 Preliminaries

We consider $n \geq 3$ disjoint disks of radius R in the Euclidean plane. We denote an optimal TSP tour on the centers of the disks as TSP^*. Similarly, $TSPN^*$ will denote an optimal TSPN tour on the disks. Our results will be with respect to a fixed TSPN tour (which we call simply $TSPN$) described by a sequence of ordered points $P_1, P_2, \ldots, P_n$ on the boundary of the disks such that the tour is a polygonal cycle with edges (P_i, P_{i+1}). Furthermore, we have that for each of the input disks, there exists some $i \in [1, n]$ such that point P_i is on the boundary of the disk.

Notice that the points P_i induce a natural order σ on the disks with centers $O_1, O_2, \ldots O_n$, i.e. σ corresponds to the identity permutation on $P_1, \ldots, P_n$. For the majority of our theorems, we will assume that TSP always refers to a tour *on the centers* and in the order σ on the disks. When we need to make a difference, we will further use $TSP(\sigma')$ to be the tour which visits the centers in the order given by the permutation σ'. Given two such permutations σ and σ', we say that

$\sigma \cap \sigma'$ refers to the maximal set of points on which σ and σ' agree. In this context, $TSPN(\sigma \cap \sigma')$ refers to the collection of paths we get from visiting the points P_i according to $\sigma \cap \sigma'$. Similarly, $TSP(\sigma \cap \sigma')$ corresponds to the collection of paths that we get from visiting the points O_i according to $\sigma \cap \sigma'$.

Finally, we denote the length of a tour $\mathcal{T}$ as $|\mathcal{T}|$. When $\mathcal{T}$ is a collection of paths, we have that $|\mathcal{T}|$ represents the total length of each of the paths. When A and B are points, we have that $|AB|$ denotes the length of the segment AB. We therefore have that $|TSPN| = \sum_{i=1}^{n} |P_iP_{i+1}|$ and $|TSP| = \sum_{i=1}^{n} |O_iO_{i+1}|$, where $P_{n+1} = P_1$ and $O_{n+1} = O_1$.

4.3 β-triads and a Structural Theorem

Before we formally define what a "bad" edge is, we will describe how to interpret the $2Rn$ *detour bound* from [68] as charging edges instead of vertices. We fix an order σ and consider the points P_i and O_i as previously defined. The argument in [68] then says that we must have:

$$\sum_i |O_iO_{i+1}| \leq \sum_i |P_iP_{i+1}| + 2Rn.$$

In this context, the term $\sum_i P_iP_{i+1} + 2Rn$ is the length of a tour that follows the TSPN tour and additionally, at each point P_i, takes a detour of $2R$ to visit the center O_i and come back. Choosing σ to be the optimal order in which $TSPN^*$ visits the disks gives us (1.1). In this view, the detour term $2Rn$ is obtained by charging $2R$ to each point P_i of the TSPN tour. Instead, we can also think of it as coming from charging each *edge* P_iP_{i+1} of the tour with a *local detour* of $2R$ in the

following sense:

$$|O_iO_{i+1}| \leq |P_iP_{i+1}| + 2R.$$

This new perspective is quite natural since it captures the observation that the shortest edge which visits the disks centered at O_i and O_{i+1} has length exactly $|O_iO_{i+1}| - 2R$ and hence the $TSPN$ tour has to pay at least that for each pair of consecutive disks it visits. In this sense, we decompose the global detour term of $2Rn$ into n local detour terms $|O_iO_{i+1}| - |P_iP_{i+1}|$ that essentially quantify how efficient the TSPN on the disks is locally.

In this context, saying a TSPN edge has a high local detour is equivalent to saying that it is close to being locally optimal or shortest possible: when the edge is exactly of length $|O_iO_{i+1}| - 2R$, its local detour is $2R$ (the maximum). If, on the other hand, we know that the edge is bounded away from $|O_iO_{i+1}| - 2R$, i.e. $|P_iP_{i+1}| > |O_iO_{i+1}| - 2R + \epsilon R$, for some $\epsilon > 0$, this translates into a local detour of at most $(2 - \epsilon)R$. Intuitively, such an edge is "good" for us because it allows us to lower the overall detour term. In contrast, "bad" P_iP_{i+1} edges are the ones for which the local detour term is large and consequently, their length is closer to $|O_iO_{i+1}| - 2R$. Our technique is motivated by trying to describe the behavior of such bad edges.

Formally, we consider a fixed angle parameter $\beta \in [0, \pi/12]$ that we instantiate later when we derive the overall bounds. We define the function:

$$f(O_1O_2, \beta) = \sqrt{|O_1O_2|^2 + R^2 - 2R|O_1O_2| \cos \beta},$$

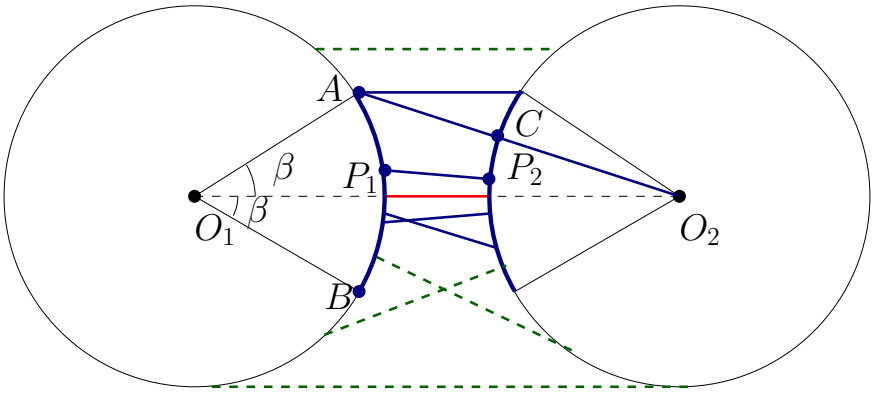


Figure 4.1: Bad edges are guaranteed to have both endpoints in the blue arcs: if $|P_1P_2| \leq |AC|$, then P_1P_2 is a bad edge. In contrast, the dashed edges are guaranteed to be good edges.

which is $|O_1O_2| - R$ when $\beta = 0$ and $|O_1O_2| + R$ when $\beta = \pi$. Intuitively, the quantity $f(O_1O_2, \beta) - R$ will control how close we are to $|O_1O_2| - 2R$. We then say that the edge P_1P_2 is **bad** if $|P_1P_2| \leq f(O_1O_2, \beta) - R$ and **good** otherwise (we abstract away the dependency on β for simplicity). Bad edges are close to $|O_1O_2| - 2R$ and will incur a large local detour. In contrast, using straightforward algebra, one can show that a good edge P_1P_2 is guaranteed to have a small detour: $|O_1O_2| \leq |P_1P_2| + (1 + \cos\beta)R$.

4.3.1 Consecutive Bad Edges

The idea behind defining bad edges in terms of $f(O_1O_2, \beta) - R$ is that it allows us to restrict the location of P_1 and P_2 on the boundary of their respective disks as seen in Figure 4.1. Specifically, there are exactly two points A and B on the boundary of the first disk with the property that the shortest distance from A or B to the boundary of the second disk is exactly $f(O_1O_2, \beta) - R$. Not coincidentally, they form an angle of β with O_1O_2 : $\angle AO_1O_2 = \angle BO_1O_2 = \beta$. In general, P_1

(and in a similar fashion P_2) is guaranteed to lie in the short arc between A and B whenever P_1P_2 is upper bounded by $f(O_1O_2, \beta) - R$:

Lemma 4.1. *If P_1P_2 is bad, then the angles $\angle O_1O_2P_2$ and $\angle O_2O_1P_1$ are $\leq \beta$.*

Proof. Let $\gamma = \angle O_1O_2P_2 \leq \pi$ and notice that $O_1P_2 = f(O_1O_2, \gamma)$. Consider the point Q where O_1P_2 intersects the first disk and note that the shortest distance from P_2 to the first disk is exactly $P_2Q = f(O_1O_2, \gamma) - R$. We therefore get that $P_1P_2 \geq P_2Q$. Now notice that, if $\gamma > \beta$, then $f(O_1O_2, \gamma) > f(O_1O_2, \beta)$ and so $P_2Q > f(O_1O_2, \beta) - R$, which would lead to a contradiction. The same argument can be applied for P_2 and we get our conclusion. □

When a second bad edge P_2P_3 is considered, we can conclude that the angle $O_1O_2O_3$ has to be at most 2β and hence the TSP on the centers must make a sharp turn after it visits O_2. Specifically, let O_3 be the center of the disk visited next at P_3 and assume that the edge P_2P_3 is also bad. Notice that the angle $\angle O_1O_2O_3$ formed by the TSP is either $\angle O_1O_2P_2 + \angle P_2O_2O_3$ or $|\angle O_1O_2P_2 - \angle P_2O_2O_3|$. Regardless, we have that $\angle O_1O_2O_3 \leq \angle O_1O_2P_2 + \angle P_2O_2O_3$ and get the following corollary:

Corollary 4.1. *If both P_1P_2 and P_2P_3 are bad edges, then the angle $\angle O_1O_2O_3$ is $\leq 2\beta$.*

If that happens and the disks are close to each other, we have that one of the edges of the TSP must actually intersect a disk twice. Specifically, if $|O_1O_2| \leq R/\sin(2\beta)$, then the support line for O_2O_3 must pass through the disk centered at O_1. In general, it is not true that if O_2O_3 intersects the first disk, we immediately

get that the associated TSPN edge P_2P_3 must also intersect it. In our case, however, we have that the slope of P_2P_3 is very close to the one of O_2O_3 due to the fact that it is a bad edge. We use this information to show that if O_2O_3 does not intersect the first disk, then P_2P_3 cannot be a bad edge.

Theorem 4.2. *If P_1P_2 and P_2P_3 are bad edges and $O_1O_2 \leq R/\sin(2\beta)$, then the segment P_2P_3 intersects the disk centered at O_1.*

Proof. We consider the case in which $\angle O_1O_2O_3 = \angle O_1O_2P_2 + \angle P_2O_2O_3$ and note that all the other cases are similar. We denote the two lines originating at O_2 that are tangent to the first circle as ℓ_1 and ℓ_2 such that the line O_2O_3 is in between ℓ_1 and O_2O_1. Note that this is possible because the angle that ℓ_1 forms with O_2O_1 is at least 2β (since $O_1O_2 \leq R/\sin(2\beta)$) but the angle that O_2O_3 forms with O_2O_1 is at most 2β (Corollary 4.1).

Our strategy will be to first show that the segment P_2P_3 is contained in the wedge defined by ℓ_1 and ℓ_2 (Figure 4.2). Notice that, since the wedge defines a convex space, it is enough to show that P_2 and P_3 are contained in it.

We first show that the point P_2 has to be in the wedge. Let S_1 and S_2 be the points in which the segment O_1O_2 intersects the first and second disk. Similarly, let T_2 and T_3 be the points in which O_2O_3 intersects the second and third disk. We then have that P_2 is between T_2 and S_2.

Now we only need to show that P_3 is in between ℓ_1 and ℓ_2. We will do that by arguing that any choice of P_3 outside of the wedge will contradict the fact that P_2P_3 is a bad edge.

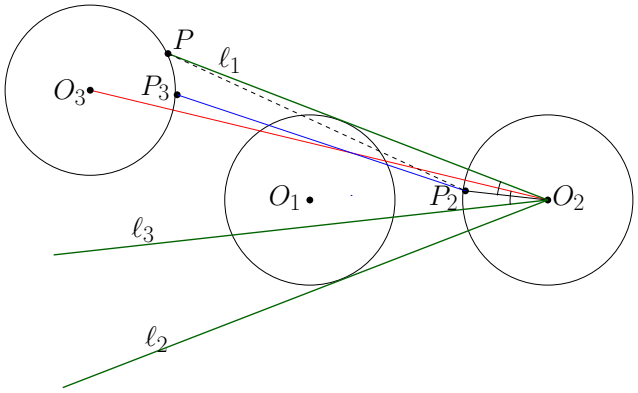


Figure 4.2: When O_2O_3 crosses the disk centered at O_1, we must also have that the segment P_2P_3 also crosses it. We show this by arguing that P_2P_3 is contained between the two lines ℓ_1 and ℓ_3 and that P_2 and P_3 are on separate sides of the first disk.

Let $\alpha_1 = \angle O_1O_2P_2$ and $\alpha_2 = \angle P_2O_2O_3$, and so $\alpha_1, \alpha_2 \leq \beta$ (Lemma 4.1). First notice that if neither ℓ_1 nor ℓ_2 intersect the third disk, then we are done because we have that the entire boundary is contained in the convex space (since O_3 is already in between ℓ_1 and ℓ_2). Assume then that ℓ_1 intersects the third disk at a point P above the line O_2O_3, since $\angle O_3O_2O_1 \leq 2\beta \leq \angle PO_2O_1$. Moreover, since $\angle PO_2O_1 \geq 2\beta$, we have that $\angle PO_2P_2 \geq 2\beta - \alpha_1 \geq \beta$ and so $PP_2 \geq f(PO_2, \beta)$ (because $|P_2O_2| = R$). Since $|PO_2| \geq |T_3O_2|$ and $|T_3O_2| \geq R$, this implies that $|PP_2| \geq f(T_3O_2, \beta)$. Using the fact that $f(x, \beta) \geq x - R\cos\beta$ for any x and $\beta \neq 0$, one can verify that:

$$\begin{aligned} f(T_3O_2, \beta) &= f(O_2O_3 - R, \beta) \\ &= \sqrt{(|O_2O_3| - R)^2 + R^2 - 2R(|O_2O_3| - R)\cos\beta} \\ &> \sqrt{|O_2O_3|^2 + R^2 - 2R|O_2O_3|\cos\beta} - R \\ &> f(O_2O_3, \beta) - R. \end{aligned}$$

This means that P cannot be a possible position for P_3 because then P_2P_3 would

be too big. Moreover, any point Q "above" P (i.e. such that $\angle O_3O_2Q > \angle O_3O_2P$) would also not work as a possible position for the same reason. In other words, P_3 has to be underneath the line $PO_2 = \ell_1$.

In order to prove that P_3 is also above the line ℓ_2, we will consider an additional line ℓ_3 originating at O_2 that makes an angle of β with O_2P_2 and is underneath it. This new line makes an angle of $\beta + \alpha_2$ with O_2O_3 and since ℓ_2 makes an angle of $\geq 2\beta + \alpha_1 + \alpha_2$ with O_2O_3, we get that ℓ_3 is in between O_2O_3 and ℓ_2. In other words, if we show that P_3 is above ℓ_3, then we also get that P_3 is above ℓ_2. If ℓ_3 does not intersect the third disk, then we are done as before, so assume that it intersects it at a point Q on the boundary. Similarly as before, we have that $P_2Q = f(O_2Q, \beta) \geq f(O_2T_3, \beta) > f(O_2O_3, \beta) - R$. This in turn implies that P_3 has to be above Q, otherwise P_2P_3 would be too big. Therefore P_3 must be above the line ℓ_3.

At this point, we have that the segment P_2P_3 is contained in the wedge defined by ℓ_1 and ℓ_2. We know that the first disk is tangent on both sides to ℓ_1 and ℓ_2 but this does not directly imply that P_2P_3 must actually intersect it. In order to have that, we must also ensure that P_2 and P_3 lie on different sides of the first disk. We argue this by showing that O_3 itself must be on the other side of the first disk as O_2. Since the disks do not intersect, this implies that P_3 is on a different side from P_2. In order to show this, notice that we can assume, without loss of generality, that $O_1O_2 \leq O_2O_3$. Let T be the point on O_2O_3 such that $O_1T \perp O_2O_3$. Since $\angle O_1O_2O_3 \leq 2\beta$ and $O_1O_2 \leq R/\sin(2\beta)$, this means that T is contained in the first disk. Suppose that O_3 is on the segment O_2T (effectively in between O_1 and O_2).

Then $O_2O3 < O_2T$ but, since $O_2T = O_1O_2\cos(\angle O_1O_2T) \leq O_1O_2$, this would lead to a contradiction. We therefore get that O_2 and O_3 are on different sides of the first disk and that the same is true for P_2 and P_3. This shows that the segment P_2P_3 must intersect the first disk. □

4.3.2 Introducing β-triads

The fact that the disk centered at O_1 is crossed by both P_1P_2 and P_2P_3 suggests that the TSPN might not be optimal because it could be shortcut. Our structural theorem identifies when that is the case and isolates the remainder as having a specialized local structure which we call a β-triad. Formally, we say that a specific TSPN subpath $P_n - P_1 - P_2 - P_3$ is a **β-triad** if it satisfies all of the following properties (Figure 4.3):

- P_1P_2 and P_2P_3 are bad edges and $O_1O_2 \leq R/\sin(2\beta)$,
- P_1, P_2, P_3 are not collinear but P_n, P_1, P_2 are collinear with P_1 between P_n and P_2.

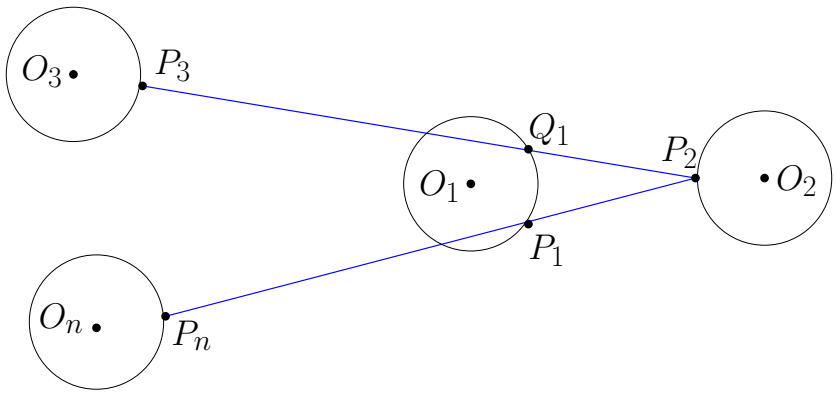


Figure 4.3: The path $P_n - P_1 - P_2 - P_3$ forms a β-triad.

We state the structural theorem here and refer the reader to Appendix **??** for a complete argument. The case in which the TSPN tour follows a straight line that stabs all the disks is discussed separately in Section BLAH BLAH BLAH and is of separate interest.

Theorem 4.3. *For $n \geq 4$, if P_1P_2 and P_2P_3 are bad edges and $O_1O_2 \leq R/\sin(2\beta)$ then at least one of the following is true:*

- *the TSPN tour is not optimal,*
- *the TSPN tour is supported by a straight line or*
- *the path $P_n - P_1 - P_2 - P_3$ forms a β-triad.*

Proof. We distinguish between the case in which P_2P_3 intersects the first disk at P_1 and otherwise. In the first case, we will show that either the TSPN is not optimal or all the disks are stabbed by it. The second case is more involved and reduces to describing what the local structure of the TSPN must be such that it does not necessarily fall in the previous two cases.

Case 1: P_1, P_2, P_3 are collinear. Then consider the point P_n that connects to P_1. The cost that the TSPN pays for visiting the four disks is $|P_nP_1|+|P_1P_2|+|P_2P_3|$ but by triangle inequality, we know that $|P_nP_2| \leq |P_nP_1|+|P_1P_2|$, so the TSPN would visit P_2 directly and pass through P_1 on its way to P_3. If the inequality is strict, then this directly implies that the TSPN is not optimal. When we have equality, however, this implies that P_n, P_1 and P_2 are now also collinear and furthermore, that P_1 lies between P_2 and P_n. In other words, we have that on the line from P_2 to P_n, we have

both P_3 and P_n to the left of P_1. Now look at how point P_4 connects to P_3 and notice that the portion of TSPN for the five disks is now $|P_4P_3|+|P_3P_2|+|P_1P_2|+|P_1P_n|$ and again, we can ask the question of why wouldn't the TSPN go straight to P_2 instead and visit P_3 along the line P_2P_3. Specifically, we have $|P_4P_2| \leq |P_4P_3| + |P_3P_2|$ with the TSPN not being optimal whenever this inequality is strict. We therefore consider the case in which $|P_4P_2| = |P_4P_3| + |P_3P_2|$ and get that now P_4 has to also be collinear with the other points and furthermore, P_3 has to be between P_4 and P_2. Continuing this process, we get that all the TSPN points would have to be collinear and in the order $P_2, P_1, P_3, P_4, \ldots, P_{n-1}$ with P_n potentially being anywhere past P_1. In this case, we have that the TSPN is a supported by a straight line that stabs all of the disks.

Case 2: P_1, P_2, P_3 are not collinear. Let the line P_1P_2 intersect the first disk for the first time at Q_1. By the argument from before, we know that if $|P_nP_2| < |P_nP_1| + |P_1P_2|$, then the TSPN cannot be optimal since another tour could go from P_n straight to visiting P_2 and then visit P_1 on the way to P_3, at a lesser cost. When P_n, P_1 and P_2 are collinear, in that order, we say that $P_n - P_1 - P_2 - P_3$ form a β-triad.

□

4.3.3 Properties of β-triads

Theorem 4.3 says that if $|TSPN^*|$ is not a straight line, then the triad has a local detour of at most $3\sqrt{3}R$. Lemma 4.2 further states that all the bad triads are

also edge disjoint. In order to prove that, we go back to the proof of Theorem 4.3. Note that we distinguished between the case in which P_1, P_2 and P_3 are collinear (Case 1) and when they are not(Case 2). The first case leads to the TSPN being a straight line, which is ruled out by our assumptions. In the second case, the optimality of TSN^* implies that P_n, P_1 and P_2 are also collinear, with P_1 between P_2 and P_n.

Lemma 4.2. *All the β-triads in a given TSPN tour are edge disjoint.*

Proof. Assume there is another bad triad that shares edges with $P_n - P_1 - P_2 - P_3$. We distinguish four cases, based on the type of edges they have in common.

Case 1: $P_{n-1} - P_n - P_1 - P_2$ is a bad triad. This case cannot happen since P_n, P_1, P_2 are collinear.

Case 2: $P_{n-2} - P_{n-1} - P_n - P_1$ is a bad triad. Then, by definition, we must have that P_nP_1 is also a bad edge. We will show, however, than this cannot be. For this, we will use an additional lemma:

Lemma 4.3. *If P_iP_{i+2} is a straight line that passes through point P_{i+1} such that P_{i+1} is between P_i and P_{i+2}, then it cannot be that both P_iP_{i+1} and $P_{i+2}P_{i+1}$ are bad edges.*

Proof. Assume that both P_iP_{i+1} and $P_{i+2}P_{i+1}$ are bad edges. Then Corollary 4.1 implies that the angle $\angle O_iO_{i+1}O_{i+2} \leq 2\beta$. Now consider the convex hull of the two disks centered at O_i and O_{i+1} (Figure 4.4). If the disk centered at O_{i+2} intersects the convex hull, then P_{i+2} must be contained in that convex hull, otherwise the line

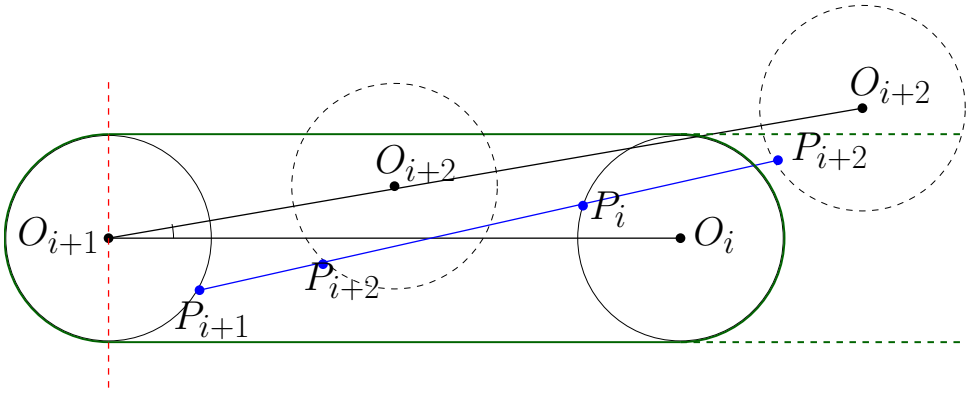


Figure 4.4: A potential TSPN path is drawn in blue. Because the angle $\angle O_i O_{i+1} O_{i+2} \leq \pi/2$, we have that O_i and O_{i+1} are on the same side of the hyperplane described by the red line. That, in turns, give two options for the disk centered at O_{i+2} to intersect the (extended) convex hulls of the other two diskss, drawn in green. In each case, the points are visited in the wrong order.

$P_i - P_{i+1} - P_{i+2}$ would not exist. But in that case, the points would be visited out of order. Specifically, P_{i+2} would be between P_i and P_{i+1}.

Now assume that the disk centered at O_{i+2} does not intersect the convex hull. Since the angle $\angle O_i O_{i+1} O_{i+2} \leq 2\beta \leq \pi/6$, this implies that O_{i+2} is in the same halfspace as O_i with respect to the line perpendicular to $O_i O_{i+1}$ passing through O_{i+1}. We extend the convex hull infinitely in that halfspace by allowing the tangent lines to be infinite on that side. By the same argument as before, we know that the disk centered at O_{i+2} must intersect this extended region. But then we would get again that the points are out of order: P_i would be between P_{i+1} and P_{i+2}. □

When $P_i = P_n, P_{i+1} = P_1$ and $P_{i+2} = P_2$, Lemma 4.3 tells us that it cannot be that P_1 is between P_n and P_2 and both edges $P_1 P_n$ and $P_1 P_2$ are bad. Therefore we are done with this case.

Case 3: $P_1 - P_2 - P_3 - P_4$ is a bad triad. This case is similar to Case 1 and cannot happen, since P_1, P_2 and P_3 cannot be collinear. **Case 4: $P_2 - P_3 - P_4 - P_5$**

is a bad triad. This case is similar to Case 2 because we have that P_2, P_3 and P_4 are collinear with P_3 between P_2 and P_4 and both P_2P_3 and P_3P_4 being bad edges.

□

4.3.4 Structural Theorem

The case of β-triads is interesting because it arises naturally as a consequence of dealing with regions instead of points. Given any optimal tour that exhibits internal angles $\leq \pi/6$, we can always add an extra disk at each sharp turn that will maintain optimality, pairwise disjointness and be intersected twice by this tour, giving rise to a β-triad. It is therefore important that we understand their behaviour.

Because of the fact that P_1P_2 and P_2P_3 are bad edges, the β-triad is likely to have a high detour with respect to $TSP(\sigma)$. Nevertheless, we show that there exists an alternate ordering σ' such that the average detour of the three edges in the β-triad with respect to $(|TSP(\sigma)| + |TSP(\sigma')|)/2$ is $3\sqrt{3}R$. The order σ' takes advantage of the fact that the disk centered at O_1 is crossed twice and inverts the order in which it is visited without changing the cost of the underlying TSPN tour. The $3\sqrt{3}R$ bound comes from proving the Häme, Hyytiä and Hakula conjecture for $n = 3$ (Section 4.5). In order to be able to construct σ' consistently across multiple β-triads, we also show that β-triads are isolated events and specifically that they are edge-disjoint (Lemma 4.2).

Theorem 4.4. *If the TSPN in the order σ has k β-triads that together cover a set of edges of total length L_T, then we can construct another order σ' that agrees with*

σ on everything except the order inside the β-triads such that:

$$\frac{|TSP(\sigma)| + |TSP(\sigma')|}{2} \leq |TSP(\sigma \cap \sigma')| + L_T + 3\sqrt{3}Rk.$$

Proof. We discuss the case for $k = 1$ and show how to modify the argument for $k > 1$. Suppose that $P_n - P_1 - P_2 - P_3$ form a β-triad. We know that $TSP(\sigma)$ visits the centers of each disk in the order $O_1, O_2, O_3, \ldots O_n$. We consider an additional order σ' such that $TSP(\sigma')$ visits the centers in the order $O_2, O_1, O_3, \ldots O_n$. Notice that both σ ad σ' agree on the order $O_3, O_4, \ldots, O_n$ and that they differ in the fact that σ visits O_1 after O_n and before O_2 and σ' visits O_1 after O_2 and before O_3. Therefore, for $T' = TSP(\sigma \cap \sigma')$, we have that $|T'| = |O_3O_4| + \ldots + |O_{n-1}O_n|$, $|TSP(\sigma)| = |T'| + |O_nO_1| + |O_1O_2| + |O_2O_3|$ and $|TSP(\sigma')| = |T'| + |O_nO_2| + |O_2O_1| + |O_1O_3|$.

On the other hand, the length of the TSPN with respect to the orders σ and σ' stays the same. The local cost of visiting $P_n - P_1 - P_2 - P_3$ is $L_T = |P_nP_1| + |P_1P_2| + |P_2P_3| = |P_nP_2| + |P_2P_3|$, since P_n, P_1 and P_2 are collinear and P_1 is between P_n and P_2. We also know that P_2P_3 intersects the disk centered at O_1 at some point Q_1 that is different from P_1 (Theorem 4.2). In other words, the TSPN that visits the points $P_n - P_1 - P_2 - P_3$ can be reimagined as visiting the points $P_n - P_2 - Q_1 - P_3$ and therefore respecting the order σ'. The local cost of crossing these edges is the same as before: $|P_nP_2| + |P_2Q_1| + |Q_1P_3| = |P_nP_2| + |P_2P_3| = L_T$.

We now apply Theorem 4.5 (the $3\sqrt{3}R$ bound for $n = 3$) on the TSP tour $O_n - O_1 - O_2$ with the TSPN tour $P_n - P_1 - P_2$ and get that:

$$\begin{aligned} |O_nO_1| + |O_1O_2| + |O_nO_2| &\leq |P_nP_1| + |P_1P_2| + |P_nP_2| + 3\sqrt{3}R \\ &\leq 2|P_2P_n| + 3\sqrt{3}R. \end{aligned}$$

On the other hand, if we consider the tour $O_1 - O_2 - O_3$ with the TSPN tour $P_2 - Q_1 - P_3$, we get that:

$$\begin{aligned}|O_1O_2| + |O_2O_3| + |O_1O_3| &\leq |Q_1P_2| + |P_2P_3| + |P_3Q_1| + 3\sqrt{3}R \\ &\leq 2|P_2P_3| + 3\sqrt{3}R.\end{aligned}$$

Combining the two inequalities and rearranging some terms gives us that:

$$\begin{aligned}|TSP(\sigma)| + |TSP(\sigma')| =& 2|T'| + |O_nO_1| + |O_1O_2| + |O_2O_3| + \\ &+ |O_nO_2| + |O_2O_1| + |O_1O_3| \\ =& 2|T'| + |O_nO_1| + |O_1O_2| + |O_nO_2| + \\ &+ |O_1O_2| + |O_1O_3| + |O_2O_3| \\ \leq& 2|T'| + 2|P_2P_n| + 2|P_2P_3| + 6\sqrt{3}R.\end{aligned}$$

Since $L_T = |P_2P_n| + |P_2P_3|$, we get our conclusion.

When $k > 1$, we construct the order σ' by switching the order in which we visit the centers in each β-triad in the same way as before. Since all the β-triads are edge disjoint (Lemma 4.2), we can construct σ' without any conflicts because any reordering that happens in one β-triad will not affect another β-triad. □

4.4 Improved Bounds on TSPN

Our main strategy will be a careful balancing of good and bad edges, in which the detour of good edges will be upper bounded by $(1+\cos\beta)R$ and that of bad edges by $2R$. While the bad edges will have the highest detour possible, we will use the fact

that they must also be large in order to lower bound the TSPN tour more efficiently than Lemma 4.5 from [68] and [143], which we quote here for completeness.

We quote the more general version formulated in [143], since we will actually use it with a slight modification.

Lemma 4.4. *[143] Given a connected geometric graph $G = (V, E)$ in $\mathbb{R}^2$ and C the set of points that are at most x away from the vertices and edges of G, we have that:*

$$Area(C) \leq 2x \cdot |G| + \pi x^2,$$

and this is tight in general.

The shape C defined above can alternatively be thought of as the shape we describe when we slide a disk of radius x along the edges of G (i.e. the Minkowski sum of G with a disk of radius x). The charging scheme behind the analysis starts at an arbitrary vertex of G and initially pays a charge of πx^2 for it. Next, each edge e we sweep from this vertex to the next will incur an additional charge of only $|e| \cdot 2x$. Because the graph G is connected, we can continue this way and sweep through the entire graph while only incurring an additional charge of $e|G| \cdot 2x$. Next, we instantiate Lemma 4.4 with G being a TSPN tour and $x = 2R$ and notice that the disk of radius $2R$ visiting a vertex P on the boundary of a disk actually covers the entire disk of radius R whose boundary P is on. Since the disks are disjoint, we get that $Area(C) \geq \pi R^2 \cdot n$ and so we have that:

Lemma 4.5. *[68, 143] For n disjoint disks of radius R, we have that any TSPN tour $\mathcal{T}$ on them satisfies:*

$$\tfrac{\pi}{4}Rn - \pi R \leq |\mathcal{T}|.$$

4.4.1 Disjoint Uniform Disks

We will now show the proof of Theorem 4.1.

Proof. Assume the $TSPN^*$ is not a straight line. We start by singling out the β-triads and considering the two orderings σ and σ' from Theorem 4.4. If there are k_1 β-triads $\mathcal{T}_1, \ldots, \mathcal{T}_{k_1}$ spanning edges of total length L_T, we get that:

$$\begin{aligned} |TSP^*| &\leq \frac{|TSP(\sigma)| + |TSP(\sigma')|}{2} \\ &\leq |TSP(\sigma \cap \sigma')| + L_T + 3\sqrt{3}R \cdot k_1. \end{aligned}$$

Observe that $TSPN(\sigma \cap \sigma')$ is a collection of disjoint paths. From all of these paths, we further extract each from these a total of k_2 subpaths $\mathcal{G}_1, \ldots, \mathcal{G}_{k_2}$ consisting of good edges. Notice that the remaining subpaths left in $\sigma \cap \sigma'$ consist of bad edges which do not form a β-triad. Suppose we obtain l such remaining subpaths paths $\mathcal{B}_1, \ldots, \mathcal{B}_l$. In other words, we have decomposed the TSPN into three categories of subpaths:

- k_1 β-triads $\mathcal{T}_1, \ldots, \mathcal{T}_{k_1}$,
- k_2 paths $\mathcal{G}_1, \ldots, \mathcal{G}_{k_2}$ that cover the remaining good edges, and
- l paths $\mathcal{B}_1, \ldots, \mathcal{B}_l$ that consist only of bad edges which do not form β-triads.

We are now ready to evaluate the detour that each of these paths takes. For each $i \in [1, k_2]$ let ψ_i the natural order on the disks associated with $\mathcal{G}_i$ and let n_i be the number of edges in $\mathcal{G}_i$. We have that:

$$|TSP(\psi_i)| \leq |TSPN(\psi_i)| + (1 + \cos\beta)R \cdot n_i.$$

When it comes to the paths $\mathcal{B}_j$, with $j \in [1, l]$, let σ_j be their natural associated orders and let m_j be the number of edges it contains. We have that $|TSP(\sigma_j)| \leq |TSPN(\sigma_j)| + 2R \cdot m_j$.

Let $N = \sum_{i=1}^{k_2} n_i$ be the total number of edges in $\mathcal{G}_1, \ldots, \mathcal{G}_{k_2}$ and $M = \sum_{j=1}^{l} m_j$ the total number of edges in $\mathcal{B}_1, \ldots, \mathcal{B}_l$. By construction, we decomposed $TSPN(\sigma \cap \sigma')$ into these two groups of edge disjoint paths and we therefore get that:

$$\begin{aligned}
|TSP(\sigma \cap \sigma')| &= \sum_{i=1}^{k_2} |TSP(\psi_i)| + \sum_{j=1}^{l} |TSP(\sigma_j)| \\
&\leq \sum_{i=1}^{k_2} \Big(|TSPN(\psi_i)| + (1 + \cos\beta)R \cdot n_i \Big) \\
&\quad + \sum_{j=1}^{l} \Big(|TSPN(\sigma_j)| + 2R \cdot m_j \Big) \\
&\leq |TSPN(\sigma \cap \sigma')| + (1 + \cos\beta)RN + 2RM.
\end{aligned}$$

Including the β-triads back into our bound, we get that:

$$\begin{aligned}
|TSP^*| &\leq \frac{|TSP(\sigma)| + |TSP(\sigma')|}{2} \\
&\leq |TSP(\sigma \cap \sigma')| + L_T + 3\sqrt{3}R \cdot k_1 \\
&\leq |TSPN(\sigma \cap \sigma')| + L_T + 3\sqrt{3}R \cdot k_1 + (1 + \cos\beta)RN + 2RM \\
&\leq |TSPN| + 3\sqrt{3}R \cdot k_1 + (1 + \cos\beta)RN + 2RM.
\end{aligned}$$

In other words, we've expressed the total detour of the $TSPN$ according to edges that participate in β-triads, edges in $\mathcal{G}_1, \ldots, \mathcal{G}_{k_2}$ and edges in $\mathcal{B}_1, \ldots, \mathcal{B}_l$. By construction, none of these paths share edges and so $3k_1 + N + M = n$. Let

$K = 3k_1 + N$ be the total number of edges either in a β-triad or in $\mathcal{G}_1, \ldots, \mathcal{G}_{k_2}$ and since $\sqrt{3} \leq 1 + \cos\beta$, we have that:

$$|TSP^*| \leq |TSPN| + (1 + \cos\beta)R \cdot K + 2R \cdot (n - K).$$

Case 1: when $K \geq \frac{n}{2}$. In this situation, we have that:

$$|TSP^*| \leq |TSPN^*| + \frac{3 + \cos\beta}{2} \cdot R \cdot n.$$

The average detour per edge $\frac{3+\cos\beta}{2}$ is better than the $2R$ bound, but it is constrained by the choice of $\beta \in [0, \pi/12]$, which means that the best we could hope for is an average detour of $\frac{1}{2}(3+\cos\frac{\pi}{12})R < 1.983R$. We note that the average detour in the Häme, Hyytiä and Hakula conjecture is $\sqrt{3}R \approx 1.732R$. Using Lemma 4.5 gives us that

$$|TSP^*| \leq \left(1 + \frac{2}{\pi} \cdot (3 + \cos\beta)\right) \cdot |TSPN^*| + \\ + 2 \cdot (3 + \cos\beta)R \quad .$$

For large n, the $1 + \frac{2}{\pi} \cdot (3 + \cos\beta)$ term will dominate our approximation factor and is at most 3.525, when $\beta = \pi/12$.

Case 2: when $K < \frac{n}{2}$. In this situation, even the overall detour might be large, we will show that in fact, in this case, TSP^* is a 2-approximation and therefore, the best that it can be in general. We know that each path $\mathcal{B}_j$ consists of bad edges which do not form any β-triads. In other words, if P_1P_2 is an edge in it, then we know that $|O_1O2| > R/\sin(2\beta)$ which in turn means that $|P_1P_2| >$

$(1/\sin(2\beta) - 2) \cdot R$. Overall we have that:

$$|TSPN| \geq \sum_{j=1}^{l} |\mathcal{B}_j| \geq \Big(\frac{1}{\sin(2\beta)} - 2\Big) R \cdot (n - K)$$
$$\geq \Big(\frac{1}{2\sin(2\beta)} - 1\Big) R \cdot n.$$

Since the total detour could be at most $2R$ per edge, we get that:

$$|TSP^*| \leq \Big(1 + \frac{2}{\frac{1}{2\sin(2\beta)} - 1}\Big) \cdot |TSPN|.$$

When $\beta = \frac{1}{2}\arcsin\frac{1}{6}$, the detour from Case 1 becomes $\frac{3+\cos\beta}{2} \approx 1.998$ and the approximation factor from Case 2 becomes exactly 2. We note that the machinery described can be used to obtain more nuanced results. In particular, lower choices for β will drive the approximation factor in Case 2 even lower than 2, at the expense of a higher detour bound for Case 1.

□

More generally, we can consider a parameter $\alpha > 1$ that we will set later in the proof. We include here only the aspects that change. Depending on whether $K \leq \frac{n}{\alpha}$ or not, we will employ different lower bounds on $|TSPN|$, in a similar fashion as before.

Case 1: when $K \geq \frac{n}{\alpha}$. In this situation, we have that:

$$|TSP^*| \leq |TSPN^*| + \frac{1 + \cos\beta + 2(\alpha - 1)}{\alpha} \cdot R \cdot n.$$

Using Lemma 4.5 gives us that

$$\begin{aligned}|TSP^*| &\leq \Big(1+\frac{4}{\pi}\cdot\frac{1+\cos\beta+2(\alpha-1)}{\alpha}\Big)\cdot|TSPN^*|+4\cdot\frac{1+\cos\beta+2(\alpha-1)}{\alpha}R\\ &\leq \Big(1+\frac{4}{\pi}\cdot\frac{1+\cos\beta+2(\alpha-1)}{\alpha}\Big)\cdot|TSPN^*|+8R\\ &\leq \Big(1+\frac{8}{\pi}-\frac{4}{\pi}\cdot\frac{1-\cos\beta}{\alpha}\Big)\cdot|TSPN^*|+8R.\end{aligned}$$

Case 2: when $K < \frac{n}{\alpha}$. We know that each path $\mathcal{B}_j$ consists of bad edges which do not form any β-triads. In other words, if P_1P_2 is an edge in it, then we know that $|O_1O2| > R/\sin(2\beta)$ which in turn means that $|P_1P_2| > (1/\sin(2\beta)-2)\cdot R$. Overall we have that:

$$\begin{aligned}|TSPN| &\geq \sum_{j=1}^{l}|\mathcal{B}_j|\\ &\geq \Big(\frac{1}{\sin(2\beta)}-2\Big)R\cdot(n-K)\\ &\geq \frac{\alpha-1}{\alpha}\cdot\Big(\frac{1}{\sin(2\beta)}-2\Big)R\cdot n.\end{aligned}$$

Since the total detour could be at most $2R$ per edge, we get that:

$$|TSP^*| \leq \Big(1+\frac{\alpha}{\alpha-1}\cdot\frac{2}{\frac{1}{\sin(2\beta)}-2}\Big)\cdot|TSPN|.$$

If we want to achieve a factor 2-approximation in Case 2, we need to have $\beta \leq \frac{1}{2}\arcsin(\frac{1}{4})$ and set

$$\alpha = 1+2/(\tfrac{1}{\sin(2\beta)}-4).$$

In this case, the detour in Case 1 becomes $2-(1-\cos\beta)(2-1/(1-2\sin(2\beta))$ which achieves a minimum of ≈ 0.998 on the interval $[0,\frac{1}{2}\arcsin(\frac{1}{4})]$. Setting $\alpha = 1+2/(c/\sin(2\beta)-2c-2)$ for $c=2.53$ and $\beta=0.1831$ gives us that both of these cases lead to a 2.53-approximation.

4.4.2 Overlapping Uniform Disks

We discuss how the analysis from the disjoint case carries over to the case of overlapping disks. As we mentioned before, the best known approximation for this case is by Dumitrescu and Tóth [143]. In general, approaches for this case take advantage of known analyses for the disjoint case and adapt them in a smart way to the overlapping case. We begin by roughly describing the technique of Dumitrescu and Tóth [143] and the show how the analysis changes when we use our framework.

Specifically, Dumitrescu and Tóth [143] start by computing a monotone maximal set of disjoint disks $\mathcal{I}$ by greedily selecting the leftmost disk and deleting all of the other input disks that intersect it. Let k be the size of the set we end up wth. They then compute an approximate TSP tour on the centers of the disks in $\mathcal{I}$, either using the available schemes [136, 137] or Christofides [32]. We call this tour $T_{\mathcal{I}}$. They then augment this tour in such a way that we visit all the input disks, not just the ones in $\mathcal{I}$. Before we discuss the augmentation part, we first define some notation and mention some bounds that follow naturally.

Let the optimal TSP tour on the centers in $\mathcal{I}$ be $TSP^*_{\mathcal{I}}$. The eventual tour $T_{\mathcal{I}}$ that we compute will be an a-approximation to $TSPN^*_{\mathcal{I}}$ so we have that:

$$|T_{\mathcal{I}}| \leq a \cdot |TSP^*_{\mathcal{I}}|. \tag{4.1}$$

On the other hand, we know that this set of disks also has an associated optimal TSPN tour, which we call $TSPN^*_I$. Finally, we denote the optimal TSPN

tour on all the disks by $TSPN^*$. We know that the tour on $\mathcal{I}$ is a lower bound:

$$|TSPN^*_{\mathcal{I}}| \leq |TSPN^*|. \tag{4.2}$$

The size of our final solution will be compared to $|TSPN^*|$ and to that end, we use lower bounds on $|TSPN^*_{\mathcal{I}}|$ in conjunction with (4.2) to get lower bounds on $|TSPN^*|$. This is the part where our new framework will come in, because $|TSPN^*_{\mathcal{I}}|$ is a tour on disjoint disks by definition.

The next step is to augment $T_{\mathcal{I}}$ with detours of length $O(R)$ along the disks in $\mathcal{I}$ such that it touches every other disk not in $\mathcal{I}$. The total length of the solution would then become $|\mathcal{T}_{\mathcal{I}}| + O(1) \cdot |\mathcal{I}| \cdot R$. Specifically, Dumitrescu and Tóth [143] consider short curves around each disk in $\mathcal{I}$ that are guaranteed to cross any of the disks to its right that intersect it. Because the maximal set was chosen from left to right, that covers all the disks that could possibly intersect it. We refer the reader to [143] for the detailed construction. The authors show that the length of the resulting tour T is within $O(1) \cdot |\mathcal{I}| \cdot R$ of $|T_{\mathcal{I}}|$:

$$|T| \leq |T_{\mathcal{I}}| + (A \cdot k + B) \cdot R, \tag{4.3}$$

where $A = 2 \cdot (\frac{\pi}{6} + \sqrt{3} - 1)$ and $B = 4 - \sqrt{3}$.

Combining 4.1 and 4.3, we upper bound the length of the solution $|T|$ in terms of $|TSP^*_{\mathcal{I}}|$ as such:

$$\begin{aligned} |T| &\leq |T_{\mathcal{I}}| + (A \cdot k + B) \cdot R \\ &\leq a \cdot |TSP^*_{\mathcal{I}}| + (A \cdot k + B) \cdot R \end{aligned}$$

In order to complete the analysis, we would need to bound $|TSP^*_{\mathcal{I}}|$ in terms of $|TSPN^*|$ and we do that through $|TSPN^*_{\mathcal{I}}|$. The analysis from Dumitrescu and Tóth [143] uses the bounds from Dumitrescu and Mitchell [68] for the case of disjoint disks. Specifically, they apply Lemma 4.5 to get that:

$$kR \leq \frac{4}{\pi} \cdot |TSPN^*_{\mathcal{I}}| + 4R.$$

This, together with the bound $|TSP^*_{\mathcal{I}}| \leq |TSPN^*_{\mathcal{I}}| + 2Rk$ and (4.2) yields:

$$\begin{aligned}
|T| &\leq a \cdot |TSP^*_{\mathcal{I}}| + (Ak + B) \cdot R \\
&\leq a \cdot (|TSPN^*_{\mathcal{I}}| + 2Rk) + (Ak + B) \cdot R \\
&\leq a \cdot |TSPN^*_{\mathcal{I}}| + (2a + A) \cdot kR + BR \\
&\leq a \cdot |TSPN^*_{\mathcal{I}}| + (2a + A) \cdot \left(\frac{4}{\pi}|TSPN^*_{\mathcal{I}}| + 4R\right) + BR \\
&\leq \left(a + (2a + A)\frac{4}{\pi}\right) \cdot |TSPN^*_{\mathcal{I}}| + (8a + 4A + B)R \\
&\leq \left((1 + \frac{8}{\pi})a + \frac{4A}{\pi}\right) \cdot |TSPN^*_{\mathcal{I}}| + (8a + 4A + B)R \\
&\leq \left((1 + \frac{8}{\pi})a + \frac{4A}{\pi}\right) \cdot |TSPN^*| + (8a + 4A + B)R
\end{aligned}$$

Plugging in the values for A and B gives an overall approximation term of:

$$(1 + \frac{8}{\pi})a + \frac{4A}{\pi} \leq \left(\frac{7}{3} + \frac{8\sqrt{3}}{\pi}\right) \cdot (1 + \epsilon) \leq 6.75 \cdot (1 + \epsilon).$$

Our framework changes the last stage in which we compare $|TSP^*_{\mathcal{I}}|$ with $|TSPN^*_{\mathcal{I}}|$. We do a similar analysis as in the disjoint case, except for the tour on $\mathcal{I}$. We get that **Case 1** would therefore correspond to getting that:

$$|TSP^*_{\mathcal{I}}| \leq |TSPN^*_{\mathcal{I}}| + X \cdot R \cdot k,$$

where $X = 2 - \dfrac{1-\cos\beta}{\alpha}$ (instead of $2R$). We can then replace it in the analysis and get:

$$\begin{aligned}
|T| &\leq a \cdot (|TSPN^*_{\mathcal{I}}| + XRk) + (Ak + B) \cdot R \\
&\leq \left(a + (Xa + A)\frac{4}{\pi}\right) \cdot |TSPN^*_{\mathcal{I}}| + (8a + 4A + B)R \\
&\leq \left((1 + \frac{4X}{\pi})a + \frac{4A}{\pi}\right) \cdot |TSPN^*| + (4Xa + 4A + B)R
\end{aligned}$$

In **Case** 2, we have that the overall detour is $2Rk$, but there is a different lower bound on $|TSPN^*_{\mathcal{I}}|$:

$$|TSPN^*_{\wr}| \geq Y \cdot Rk,$$

where $Y = \frac{\alpha-1}{\alpha} \cdot \Big(1/(2\sin(2\beta)) - 1\Big)$. Using the fact that $Rk \leq 1/Y \cdot |TSPN^*_{\mathcal{I}}|$, the analysis then becomes:

$$\begin{aligned}
|T| &\leq \alpha \cdot |TSPN^*_{\mathcal{I}}| + (2\alpha + A) \cdot kR + BR \\
&\leq \alpha \cdot |TSPN^*_{\mathcal{I}}| + \frac{2\alpha + A}{Y} \cdot |TSPN^*_{\mathcal{I}}| + BR \\
&\leq \left(\alpha + \frac{2\alpha + A}{Y}\right) \cdot |TSPN^*_{\mathcal{I}}| + BR.
\end{aligned}$$

If we set α and β like in the previous section, we get that both of the approximation factors are upper bounded by 6.728.

4.4.3 The Straight Line Case

Here we focus on the second possibility in Theorem 4.3 in which the optimal TSPN is supported by a straight line that stabs all the disks. We show that in this case, we can return in polynomial time a solution that is within an additive factor of $4R$ from the optimal $TSPN^*$. We note that when the TSPN might not be a line

but the disks themselves admit a line transversal, a $\sqrt{2}$-approximation follows from the work of Dumitrescu and Mitchell [68]. We explain the result for completeness.

We start by identifying the centers that are the farthest apart and considering the direction orthogonal to the line going through them. This direction induces parallel segments of length $2R$ in each of the disks (that each go through the centers). It is easy to check that any line transversal through the disks is a line transversal through the segments except for the first and last disk in the associated geometric permutation (for those two disks, the TSPN will stop at the boundary of the disk and never cross the entire circle). Conversely, any line transversal through the segments will automatically also stab the disks. Now compute a shortest line segment that stabs all of these segments in time $O(n \log n)$ using the algorithm of Bhattacharya et al. [148]. We note that this is optimal up to an additive factor of $4R$ that comes from the fact that the optimal $TSPN^*$ might have to travel $4R$ to hit the first and the last two segments in the geometric permutation.

In general, when we know that the disks admit a line transversal, we can output a solution that is a $\sqrt{2}$-approximation [68]. This follows indirectly from an algorithm used for connected regions of the same diameter, when there is a line that stabs all of the diameters. Given the parallel segments of length $2R$ that we constructed earlier, we know that they can also be stabbed by a line. Now consider the smallest perimeter axis-aligned rectangle that intersects all of the segments, of width w and height h. This will be the solution that we return. Arkin and Hassin [70] argued that any tour which touches all four sides of the rectangle must have length at least $2\sqrt{h^2 + w^2}$. Since $h + w \leq \sqrt{2} \cdot \sqrt{h^2 + w^2}$, we get that the

rectangle is a $\sqrt{2}$-approximation.

4.5 The Fermat-Weber Approach

In this section, we prove that the Häme, Hyytiä and Hakula conjecture is true for $n = 3$ and discuss a different way of looking at the TSPN tour that we believe might be of independent interest. We start with the observation that the shortest tour on the centers is equivalent to the shortest tour on translates of those centers, as long as all those centers are translated according to the same vector. In other words, if we fix a direction and translate each center along that direction until it reaches its boundary, the shortest tour on the newly obtained points will be exactly the same as the shortest tour on the centers themselves.

Formally, let B_i be the point we obtain by translating the center O_i along a fixed vector of length R. Then the TSP on the points $B_1, B_2, \ldots, B_n$ (in that order) has the same length as the TSP tour on $O_1, O_2, \ldots O_n$ (Figure 4.5). One advantage of visiting the first set of points (instead of the center points) is that it might be more similar geometrically to what the TSPN actually does. In terms of the following analysis, we would get that:

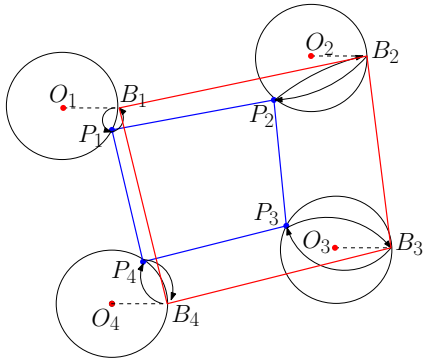


Figure 4.5: The translated view, when the tour visits the same point on the boundary of each disk.

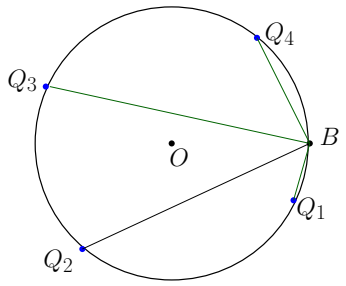
$$|TSP^*| \leq |TSPN^*| + 2 \cdot \sum_{i=1}^{n} |P_i B_i|.$$

In this context, a natural question arises about the choice for the points B_i that minimizes the term $\sum_{i=1}^{n} |P_iB_i|$. In order to see what this best choice would be, we transform this input instance into another one by essentially superimposing all the disks on top of each other (Figure 4.6(a)). Specifically, our new instance will consist of one disk of radius R centered at a point O such that the points B_i map to a single point B (corresponding to O translated by the same fixed vector). We then map each point P_i of the TSPN to a corresponding point Q_i on the boundary of this disk such the vector OQ_i is a translate of the vector O_iP_i. We then get that:

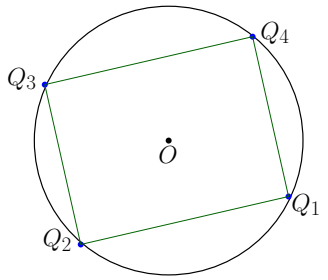
$$\sum_{i=1}^{n} |P_iB_i| = \sum_{i=1}^{n} |Q_iB|,$$

and so the best choice for B is the one that minimizes the sum $\sum_{i=1}^{n} |Q_iB|$, otherwise know as the *Fermat-Weber point* or 1-*median* of the points $Q_1, Q_2, \ldots, Q_n$ [149,150]. We note, however, that while the average distance to the Fermat-Weber point will never be greater than $2R$, there are instances in which this is tight. Consider, for example, the points Q_i to be the vertices of a convex $2n$-gon and notice by triangle inequality that the center of the disk is exactly their Fermat-Weber point (any other point will incur distances greater than the sum of the diagonals).

We can therefore say that when the points B_i are evenly spaced on the boundary of the disk the Fermat-Weber point is exactly the center and so we gain no improvement by moving the centers O_i towards the points B_i. It turns out, however, that the location of the points on the boundary is not as restrictive as the order in which the TSPN visits them. To see that, consider a different transformation in which we only move the centers O_i and O_{i+1} along a fixed vector. In other words, we



(a) Unified view when we translate each O_i to the same point on the boundary.



(b) n different choices for B, when we choose a different B for each pairs of points O_i and O_{i+1}

Figure 4.6: The detour bound depending on what kind of Fermat-Weber points we consider.

choose a new vector for each pair of consecutive centers and only compare $|P_iP_{i+1}|$ locally against the newly obtained segment. This does not give us an overall valid tour on the centers, but it allows us to tailor the choice of B for each two points P_i and P_{i+1}. Specifically, we would get that:

$$|O_iO_{i+1}| \leq |P_iP_i + 1| + |Q_iB| + |Q_{i+1}B|.$$

In this case, we know that any point on the segment Q_iQ_{i+1} minimizes the distances in question and so we get that:

$$|O_iO_{i+1}| \leq |P_iP_i + 1| + |Q_iQ_{i+1}| \text{ and } |TSP^*| \leq |TSPN^*| + \sum_{i=1}^{n} |Q_iQ_{i+1}|.$$

In other words, the largest detour obtained in this way is when the TSPN visits the points P_i in the order of the Maximum TSP on the associated points Q_i (Figure 4.6(b)). The case in which all the points are evenly distributed along the boundary no longer becomes that restrictive. We can still construct, however, instances for which the Max TSP is exactly $2Rn$ and that is when the points visited

are exactly diametrically opposite each other. Nevertheless, we are able to show that for $n = 3$, the detour is bounded by $3\sqrt{3}R$. Let A, B, C be any three points on the boundary of a circle of disk R centered at O. We then have that that $|AB| + |AC| + |BC| \leq 3\sqrt{3}R$ and the Häme, Hyytiä and Hakula conjecture for $n = 3$ follows:

Theorem 4.5. *For $n = 3$, we have that any tour which visits the disks in an order σ satisfies the bound*

$$|TSP(\sigma)| \leq |TSPN(\sigma)| + 3\sqrt{3}R.$$

Chapter 5

The Maximum Scatter Traveling Salesman Problem

In this chapter, we discuss the Maximum Scatter TSP problem, for which we strive to get polynomial time algorithms. We propose a method inspired by the case when all the points lie on the circle and discuss how it can be extended to more general cases. Specifically, we propose a property of the input point set that, when present, leads to a polynomial time algorithm for computing an optimal Maximum Scatter tour.

5.1 Introduction

In this section, we review existing work on the topic and discuss our contributions.

5.1.1 Related Work

The Maximum Scatter Traveling Salesman Problem was first introduced by Arkin et al. [80] who considered the general metric case and the Euclidean case of points on a line and on a circle. For the metric case, they showed a 0.5-approximation and showed that this is tight unless P=NP. For the line and circle case, they showed polynomial time algorithms and asked the question whether the factor 0.5 can be improved for the Euclidean case. The line case focuses on obtaining an exact ordering on the points that guarantees an optimal scatter, regardless of the underlying geometry. This is based on exploiting the combinatorial properties of the tour. The same approach is then extended to the circle case. In the line case, the maximum possible scatter of a tour can be determined in linear time (by exploiting a combinatorial bound on it) and the tour itself is then constructed in additional constant or linear time depending on the parity of number of points. In contrast, we present an alternative algorithm for points on a circle that we extend to work for more general scenarios.

Fekete [79] showed that the Euclidean version is NP-hard for dimensions three or higher but did not settle the case of two dimensions. The reduction is the same one used for Maximum TSP and involves embedding a grid graph on the surface of a three-dimensional sphere. As such, the smallest class of graphs for which Max Scatter TSP is NP-hard are graphs of maximum degree four in three or more dimensions.

Recently, Kozma and Mömke [82] gave a general efficient polynomial approx-

imation scheme for a general class of metrics, called constant doubling dimensions, that include the ℓ_2 metric. Additionally, they also designed polynomial time approximation schemes when the dimension is log-logarithmic[1], or when it is constant and ϵ is at least a constant times $1/\log\log n$.

Around the same time, Hoffman et al. [81] presented an algorithm for points on a regular grid that extends the line approach from Arkin et al. [80]. The underlying idea is that of exploiting the fixed solution for the line case (i.e. specific ordering regardless of the geometry) and augmenting it with another similar rule for the second dimension of the grid. In general, this is not necessarily guaranteed to obtain optimal results and in the worst case, gives a $\frac{\sqrt{10}}{5}$-approximation for some cases. For the case of grids with an odd number of columns or for quadratic grids, this is shown to indeed lead to an optimal solution.

5.1.2 Technical Challenges and Contributions

The new challenge that Maximum Scatter TSP or Maximum TSP bring into the conversation is that their optimality becomes hard to be exploited locally. Specifically, advances on the Euclidean TSP front have come from the ability to separate the input space into smaller regions for which we can compute the problem exactly and then incorporate into a dynamic program. Such is the case for the m-guillotine method employed by Mitchell [38] and the shifted dissection method used by Arora [36]. When it comes to Maximum Scatter TSP, such methods are no longer

[1]The time complexity known as log-logarithmic refers to runtimes that can be upper bounded by a constant time $\log\log n$, where n is the size of the input space.

of immediate use because the solution restricted to a small region should by definition be empty. In other words, local solutions are no longer guaranteed to lie in a small region of the input space. In our approach, we overcome this by starting with highly global structures such as a collection of cycles that span the input points. We then argue about when these cycles can be combined into a tour by making local arguments.

We propose a general framework inspired by the case when all the points lie on the circle and discuss how it can be extended to more general cases.

5.2 Preliminaries

We denote the n input points by $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$. We assume that all of our points are in $\mathbb{R}^2$ with the Euclidean metric. A tour then corresponds to an ordering $\sigma : \mathbb{N} \to \{1, \ldots, n\}$ on these points such that the tour consists of straight line segments of the form $P_{\sigma(i)}P_{\sigma(i+1)}$ for $i = \{1, \ldots, n\}$ and $\sigma(n+1) = \sigma(1)$. We denote the length of a segment PQ to be the associated ℓ_2 distance between the endpoints P and Q and we denote it as $|PQ|$. In this context, the *scatter* of specific tour and its associated order σ becomes

$$\min_i |P_{\sigma(i)}P_{\sigma(i+1)}|.$$

The Maximum Scatter TSP problem therefore asks for an order σ that maximizes the scatter of the associated tour.

The first thing we notice is that we can reduce this problem to that of finding a Hamiltonian tour in a threshold graph. Specifically, the scatter of the optimal tour

will be achieved by one of the input edges PQ. In other words, we have $\binom{n}{2}$ choices for the optimal scatter and we can try them all. Given a specific guess ℓ, we discard from our input the edges PQ of length less than ℓ (i.e. build the threshold graph) and attempt to compute a Hamiltonian tour on them. In general, a Hamiltonian tour on a graph is a tour that visits every point once. Notice that, if ℓ^* is the scatter of the optimal tour, then any guess $\ell \leq \ell^*$ should also ensure that G is Hamiltonian. We can therefore start with the length of the smallest edge and keep increasing our guess ℓ until we produce a proof that there can be no Hamiltonian tour on the associated threshold graph. At that point, we return the largest guess so far that produced a tour.

We therefore assume a guess ℓ from now on and concern ourselves only with edges of length $\geq \ell$. Specifically, we define

Definition 5.1. *Given a point $P \in \mathcal{P}$ and a fixed distance threshold ℓ, we define the reachability area of P to be the set $R(P) \subseteq \mathcal{P}$ of input points Q such that $|PQ| \geq \ell$.*

In other words, $R(P)$ consists of all the points that P can connect to, i.e. the neighbors it has in the associated threshold graph. In the Euclidean plane, this corresponds to the difference between the input set $\mathcal{P}$ and the ball of radius ℓ centered at P.

5.3 The Circle Case

In this section, we assume that all the input points lie on the boundary of a circle $\mathcal{D}$ with radius R. In this case, the reachability area of a point P behaves in

the following way:

- if $2R \leq \ell$, then $R(P)$ is the empty set for any P, because every point on the boundary of $\mathcal{D}$ is at most $2R$ away from P. This case is trivial and we do not consider it.

- if $2R > \ell$, then $R(P)$ is induced by an arc $A(P)$ on the boundary of $\mathcal{D}$ centered at an imaginary point P', the diametrically opposite point of P on the circle. Furthermore, the length of the interval is the same for all input points.

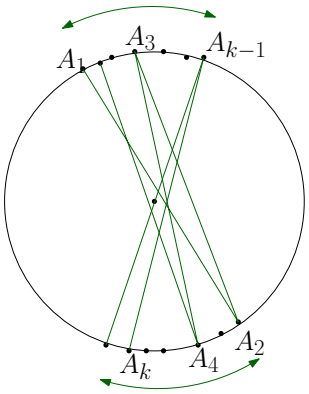
We will focus on the $2R > \ell$ scenario whenever we discuss the circle case. Our strategy for this case will be to compute a series of initial cycles and show that they can always be combined into a Hamiltonian tour unless the guess scatter ℓ is too large. We proceed to describe and identify cycles for which this is possible and conclude with a polynomial time algorithm that produces a Hamiltonian tour if one exists.

5.3.1 Description of Cycles

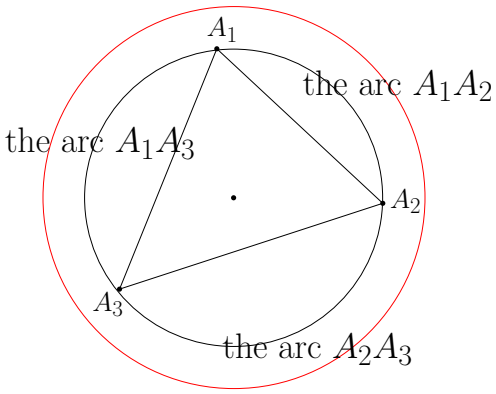
Given a cycle $\mathcal{C}$, we define the *coverage* of that cycle as being the collection of arcs that the cycle "sweeps" as it visits points. We begin by first identifying these arcs locally:

Definition 5.2. *For a subpath $A - B - C$ of a given cycle, we denote by $\widehat{AC}_B$ the arc between A and C that does not contain B.*

We can think of the arc $\widehat{AC}_B$ as capturing the fact that locally, the point B



(a) The coverage of an even cycles decomposes into two arcs (upper and lower coverage).



(b) The coverage of an odd cycle is the entire boundary of the circle.

Figure 5.1: The coverage of even versus odd cycles.

covers the area between its two neighbors in the cycle. The global coverage of the entire cycle then becomes the union of such arcs over all subpaths:

Definition 5.3. *The coverage of a cycle $\mathcal{C}$ denoted $Cov(\mathcal{C})$, is the part of the circle boundary given by:*

$$Cov(\mathcal{C}) = \bigcup_{A-B-C \in \mathcal{C}} \widehat{AC}_B.$$

The intuition behind this concept is that even cycles induce bipartite graphs and that the notion of coverage reflects this by consisting of two separate opposing intervals (Fig. 5.4(a)). Formally, we can discover the paths $A - B - C$ by ordering the vertices of the cycle. Let $P_1, P_2, \ldots, P_k$ correspond to such an ordering σ, then:

$$Cov(\mathcal{C}) = \bigcup_{i} \widehat{P_i P_{i+2} P_{i+1}}.$$

When the cycle has an even number of vertices, any ordering on them will decompose the coverage into arcs between odd indexed points and arcs between even indexed points. Specifically, we can consider the arcs between $P_1, P_3, \ldots, P_{k-1}$

and the arcs between $P_2, P_4, \ldots, P_k$. Any ordering we might consider would give rise to the same two collections and hence, we can speak of *odd/upper coverage* and of *even/lower coverage*. Specifically, given an ordering σ, we define:

$$UCov_\sigma(\mathcal{C}) = \widehat{P_1P_3} \cup \widehat{P_3P_5} \cup \ldots.$$

and

$$LCov_\sigma(\mathcal{C}) = \widehat{P_2P_4} \cup \widehat{P_4P_6} \cup \ldots.$$

Because we only consider even cycles, we get that, for any ordering σ:

$$Cov(\mathcal{C}) = UCov_\sigma(\mathcal{C}) \cup LCov_\sigma(\mathcal{C}).$$

Throughout our proofs, the choice for σ will not matter, since any ordering will give rise to the same partition of the vertices. Each of halves gives rise to a continuous arc. Moreover, each of the arcs are bounded by vertices of the cycle:

Lemma 5.1. *For any even cycle $\mathcal{C}$ that visits the points $P_1, \ldots, P_{2k}$ in that order, we have that the upper and lower coverage induce continuous arcs on the circle.*

Proof. We prove this by considering partial sums. We note that :

$$UCov_\sigma(\mathcal{C}) = \widehat{P_1P_3} \cup \widehat{P_3P_5} \cup \ldots \widehat{P_{2k-1}P_1}.$$

We start with the observation that $\widehat{P_1P_3}$ is continuous by definition. Then we consider the union $\widehat{P_1P_3} \cup \widehat{P_3P_5}$ and note that the second arc we consider overlaps with the first arc in at least one point (i.e. P_3). Since both of the components are themselves continuous, we get that their union must also be continuous. We continue like this for the rest of the sum. □

Note that the two halves are each based on disjoint set of points but they do not necessarily represent disjoint arcs on the circle. It could happen that they intersect but that will not affect our arguments.

On the other hand, if the cycle has an odd number of vertices, this partition is not necessarily consistent across orderings and we can no longer make such a distinction into upper and lower halves. In fact, we can show that the coverage of an odd cycle is not at all partitioned and must in fact cover the entire boundary of the circle (Fig. 5.4)

Lemma 5.2. *The coverage of an odd cycle is the entire boundary of the circle.*

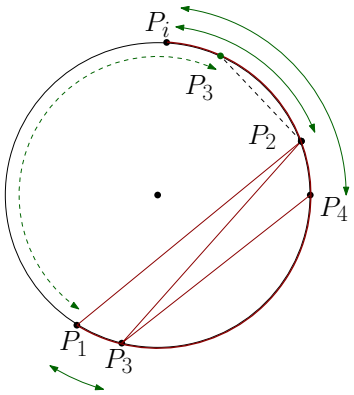


Figure 5.2: We have drawn the coverage arc Λ in dark red. Consider the possible location of P_3. In dashed green lines, we have drawn a potential location for P_3, between P_i and P_2. Notice that, in that case, the coverage arc for P_1 and P_3 would be the entire dashed green arc, because that is the one opposing P_2. Therefore, the whole arc between P_i and P_2 is restricted for P_3. Similarly, we have drawn the restricted locations for P_4 and P_5.

Proof. Assume by contradiction that the coverage does not cover the entire circle. Then there must exist an arc $\Lambda \subseteq \mathcal{D}$ bounded by vertices of the cycle that spans all its vertices: $Cov(\mathcal{C}) \subseteq \Lambda$. Let one of the boundary vertices be P_1 and P_i be those endpoints. We then get that all the points of the cycle are contained on the arc Λ

between P_1 and P_i. We then order the cycle in such a way that P_2 becomes the neighbor of P_1 that is closest to P_i. Please refer to Fig. 5.2 for a visual explanation.

We will show that there is no way for us to arrange the rest of the points such that we avoid sweeping the entire circle. In other words, we will try to restrict the location of the other points on Λ while making sure that their corresponding coverage also keeps within Λ.

First, notice that P_3 cannot be in the interval P_iP_2. If P_3 were in between P_i and P_2, then the coverage arc $\widehat{P_1P_{3P_2}}$ would include the complement of Λ, which we don't want. Let P_3 then be in the interval P_1P_2. Now consider the location of P_4. We claim that P_4 cannot be located in the interval P_1P_3. Otherwise, the coverage arc $\widehat{P_2P_{4P_3}}$ would again include the complement of Λ. Continuing this argument, we overall get that:

- P_3 cannot be between P_i and P_2
- P_4 cannot be between P_1 and P_3
- P_5 cannot be between P_i and P_4
- P_6 cannot be between P_1 and P_5 etc.
- P_{2k-1} cannot be between P_i and P_{2k-2}
- P_{2k} cannot be between P_1 and P_{2k-1}.

However, because the cycle is odd, we will eventually reach a k such that A_{2k} must actually correspond to A_1 (that way, we will have A_{2k-1} connecting back to A_1). On the other hand, our constraints say that any A_{2k} cannot be contained between

A_1 and A_{2k-1}, making it impossible for us to close the cycle while maintaining the constraint that our coverage is restricted to the arc Λ. □

We will now proceed to describe how cycles interact by looking at their coverage. Specifically, we will argue that if two cycles have overlapping coverage, then they can be combined into a bigger cycle.

5.3.2 Interleaving Cycles

Our goal is to determine cases in which two cycles spanning different point sets can be combined. Intuitively speaking, if one of the cycles connects to a point that is in the general vicinity of the second cycle, then we have identified a candidate point where a join can be made.

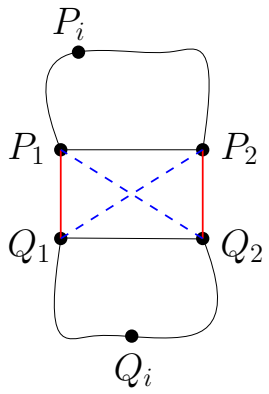


Figure 5.3: If the red edges exist, then we can form a bigger cycle by going $P_1 \dots P_i \dots P_2 - Q_2 \dots Q_i \dots Q_1 - P_2 - P_1$.

The main approach we will take for combining two cycles is that of exchanging two known edges, one from each cycle, with two newly formed edges that go across the two cycles. More specifically, we will find two adjacent points P_1, P_2 and Q_1, Q_2 in each of the cycle and claim that they have to form two disjoints connections between them (Figure 5.3). We will then cut both cycles by taking out the edges between their two neighboring points (P_1P_2 and Q_1Q_2) and glue them together using the newly formed connections (either P_1Q_1, P_2Q_2 or P_1Q_2, P_2Q_1). We begin by formally defining this strategy:

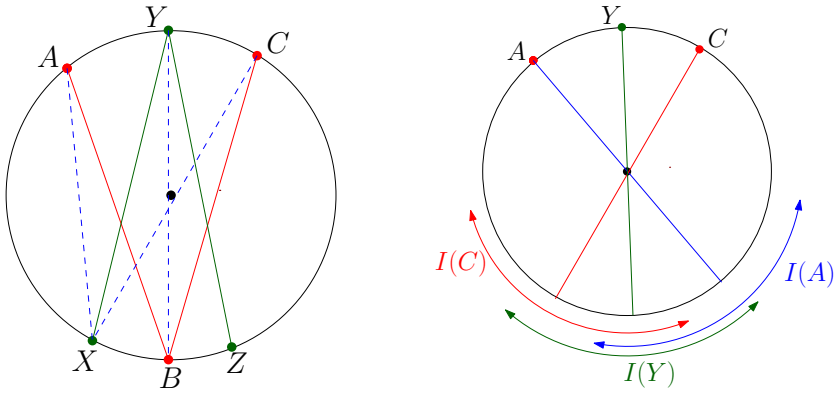
Definition 5.4. *We say that two cycles can be* ***merged*** *if we can combine them into a bigger cycle that visits all of their vertices.*

We now proceed to identify cases when such an operation can be done easily between two cycles. The first such case we discuss is when two cycles have overlapping coverages. We formally define this phenomenon as *interleaving*:

Definition 5.5. *Two cycles $\mathcal{C}_1$ and $\mathcal{C}_2$ are said to be* ***interleaving*** *if $Cov(\mathcal{C}_1) \cap Cov(\mathcal{C}_2) \neq \emptyset$. In other words, there is a subpath $A - B - C$ in $\mathcal{C}_1$ and a vertex Y in $\mathcal{C}_2$ such that $Y \in \widehat{AC}_B$. (Fig. 5.4).*

We get that:

Lemma 5.3. *Two interleaving cycles can always be merged.*



(a) We have the red $A - B - C$ cycle and the green $X - Y - Z$ cycle with the property that Y is contained in the coverage arc between A and C. We can then guarantee that the edge YB must also exist and also that AX or CX would exist.

(b) We have that $I(Y)$ is covered on the left by $I(C)$ and on the right by $I(A)$ and that $I(A) \cup I(C)$ form a continuous arc that contains $I(B)$.

Figure 5.4: Interleaving cycles always merge.

Proof. Let $A - B - C$ and $X - Y - Z$ be the subpaths as described in the definition. First, notice that since both A and C are connected to B, then Y will also be connected to B (Fig. 5.4 (a)). This is because when we move Y on the arc either towards A or C, the (acute) angle $\angle BOY$ keeps increasing, and therefore the segment BY will be longer than either BA or BC. Alternatively, from the perspective of B, the area of feasibility $I(B)$ is a continuous interval on the boundary and hence, the induced reachability area $R(B)$ consists of consecutive points. Since both A and C belong to $R(B)$, then so should Y. We identify this as a distinct property of the geometry of the points:

***Property* 1.** Any point Y on the arc $\widehat{AC}_B$ must also connect to B. In other words, $\widehat{AC}_B \subseteq I(B)$.

As an alternative proof to Property 1, we can assume that the edge YB does not exist. Then the angle $\angle BOY$ facing C would either be too small or too large. If it were too small, then it would follow that the edge BC would not exist either. If, on the other hand, the angle were too large, then its complement facing A would be too small. In that case, however, the edge AB could not exist.

Now that we have that Y and B must connect, we have an opportunity to merge the two cycles using this new connection. We show next that one of Y's neighbor in its cycle has to connect to one of B's neighbor. Since A and C share a neighbor, we have that $B \in R(A) \cap R(C)$ and therefore $I(A) \cup I(C)$ represents a continuous arc on the circle. On the other hand, since Y is in between A and C and all intervals of feasibility have the same size, we have that $I(Y)$ is a shifted version of $I(A)$ and $I(C)$ (Fig. 5.4(b)). We therefore get that $I(B)$ must be contained in

the arc $I(A) \cup I(C)$. This becomes the second useful property:

Property **2.** Any point Y on the arc $\widehat{AC}_B$ has the property that $I(Y) \subseteq I(A) \cup I(C)$.

As an alternative proof to Property 2 would be that, if we assume that the edge AZ didn't exist, we would get that the angle $\angle AOZ$ facing C would either be too small or to big. If it is too small, then the edge YZ could not exist. If, on the other hand, the angle were too large, then the complement angle facing A would be too small. Then we would get that the edge AB could not exist.

We finish the argument by considering X and Z to be Y's neighbors in the second cycle. We then have that $X, Z \in I(A) \cup I(C)$, so they are both connected to one of the vertices A or C. Notice that any one of these edges would work: AX, AZ, CX or CZ (if any of these edges exist, we can merge the two cycles). Depending on the position of X and Z relative to B, two of these edges is guaranteed to exist and we can successfully merge the two cycles. □

A direct consequence of this is that whenever we have an odd cycle present, we can always merge it with all the other cycles present. We do this by first connecting it to all the even cycles and then to the remaining odd cycles. We incorporate this as part of our algorithm later on.

5.3.3 Neighboring Even Cycles

We now consider the case in which there are no interleaving cycles and therefore, their coverage arcs are disjoint. Nevertheless, we are still able to connect

them in a meaningful way. Notice that, due to Lemma 5.2, only even cycles can be non-interleaving. From now on, these are the only cycles we will consider.

The next special case we identify is when two cycles do not interleave but for some reason still exhibit some level of connectivity:

Definition 5.6. *Two even non-interleaving cycles $\mathcal{C}_1$ and $\mathcal{C}_2$ are said to be **neighboring** if there are points A in $\mathcal{C}_1$ and X in $\mathcal{C}_2$ such that AX is an edge in the graph.*

Notice that, because the edge AX exists, we can always connect the two cycles to a path. The challenge is to do so in a strategic manner that allows us to eventually argue that they must connect as a cycle. We begin by first noticing that neighboring cycles should be able to connect at the boundary points of their coverage arcs. Specifically, given an even cycle, we have that its upper and lower halves are bounded by points of the cycle. Intuitively, if the edge AX exists, then moving A towards one of the boundary points should only increase the length of the edge AX. Formally, we get that:

Lemma 5.4. *If two even cycles are neighboring, then it must be that they can connect at their boundary points of their coverage arcs. Moreover, they will do so in a precise way that will be given by the geometry of the cycles.*

Proof. Consider one of the cycles and let E_1 and E_2 be the boundary points of the upper coverage arc and let E_3 and E_4 be the ones for the lower coverage arc (in counter-clockwise order). Suppose that A is on the upper arc and therefore sits between E_1 and E_2. Furthermore, let X belong to the second cycle and sit on the

cycle between E_1 and E_4 in counter-clockwise order (Fig. 5.5). In this case, we will show that the edge E_2X must also exist. Similar arguments can be used to treat the other cases.

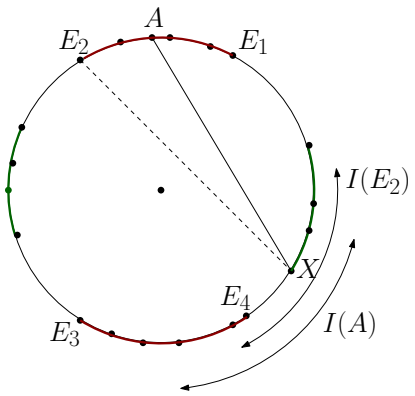


Figure 5.5: The first cycle's coverage is drawn in red and is split between the upper and the lower coverage arcs. The boundary points are E_1, E_2, E_3 and E_4. The second cycle is drawn in green.

First, notice that E_2 sits to the left of A and therefore, $I(E_2)$ will be to the right of $I(A)$ (i.e. it is shifted to the right). In other words, the right endpoint of $I(E_2)$ is to the right of the right endpoint of $I(A)$. Since $X \in I(A)$, we then have that the right endpoint of $I(E_2)$ will also be to the right of X. The reason this is the case is because E_2 has to connect to the lower half of the first cycle. Specifically, $R(E_2)$ must contain at least two points on the lower half, between E_3 and E_4. We can therefore find a point $B \in \widehat{E_3E_4}$ such that $B \in I(E_2)$. That means that $I(E_2)$ extends from B all the way to its right endpoints. Since X is in between those two points, we therefore get that X must be in $I(E_2)$. By repeating this argument from the perspective of the second cycle, we can now show that X can also be assumed to be a boundary point. Moreover, we can deduce that, if we want to determine potential edges AX, then the boundary points E_1 and E_4 will search the arc $\widehat{E_2E_{3E_1}}$

and the boundary points E_2 and E_3 will search the arc $\widehat{E_1 E_{4E_2}}$. □

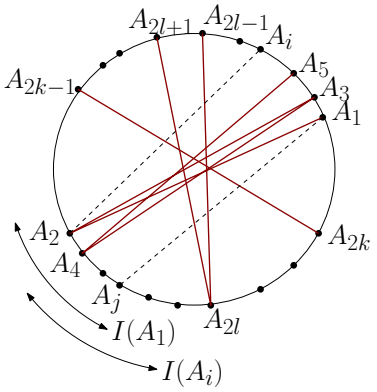
Now that we established that the boundary points of the cycles must connect, our next step is to define an ordering of the cycle that allows us to take advantage of that fact in a strategic way. Specifically, we will reduce the problem of merging two cycles to checking whether certain boundary points can connect.

Lemma 5.5. *Given an even cycle $\mathcal{C}$, we can order its vertices in such a way that they form a path that starts and ends at the corresponding left/right boundary points of the upper and lower coverage arcs. Additionally, we can construct a cycle in which opposite boundary points (from different coverage arcs) are connected.*

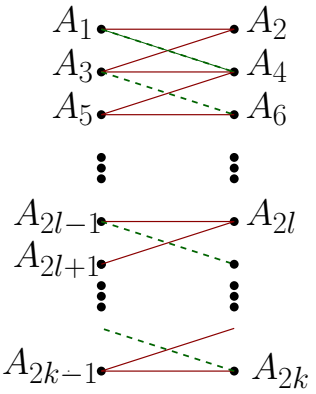
Proof. Let the cycle has $2k$ points. Order the points starting and ending at the right boundary points of the two halves, as in Fig. 5.6(a). We denote the right boundary point of the upper half as A_1 and continue in counter-clock fashion to denote points with increasing odd indices $A_3, \ldots, A_{2k-1}$ until we hit the left boundary point (A_{2k-1}). For the lower half, we denote the right boundary point as A_{2k} and continue counterclockwise in decreasing even indices until we hit the left boundary A_2.

To begin with, we first show that the vertices ordered in such a way form a path that starts at A_1 and ends at A_{2k}. As a principle, we will show that:

- A_2 must be connected to A_1 and A_3
- A_4 must be connected to A_3 and A_5
- A_{2l} must be connected to A_{2l-1} and A_{2l+1}



(a) Each even cycle can be decomposed into the odd points and the even points, that correspond to different coverage arcs. Ordering them counterclockwise gives us a path that starts and ends at the right endpoints of each coverage arc (A_1 and A_{2k}.)



(b) In red, we represented the edges that we are guaranteed to exist. Through a similar argument, we can also get that the green edges exist and therefore obtain a path that starts and ends at the left endpoints of each coverage arc (A_2 and A_{2k-1}).

Figure 5.6: Even cycles can be ordered.

This will not be the final form of the cycle, but it will help us argue about the existence of edges that will be helpful in constructing the cycle.

We begin by first showing that A_2 must be connected to A_1 and A_3. Suppose that A_1 and A_2 are not connected. Then we will argue that none of the other odd points can be connected to A_2 either. Let $A_i \neq A_1$ be an odd neighbor of A_2 in the actual cycle. Such a point will always exist because A_2 must always have at least two (odd) neighbors in the cycle. By the way we defined the points, A_i is to the left of A_1 and hence, $I(A_i)$ will be to the right of $I(A_1)$. In other words, the right endpoint of $I(A_i)$ appears after the right endpoint of $I(A_1)$ in counter-clockwise order. But since $I(A_1)$ does not include A_2 but still needs to cover other neighbors of A_1, we get that the right endpoint of $I(A_1)$ comes after A_2 in counter-clockwise

order. This in turn means that the right endpoint of $I(A_i)$ also comes after A_2 and hence that $A_2 \notin I(A_i)$. Therefore, A_1 and A_2 must be connected.

We mention here an alternative argument that depends on angle measurements. If A_1 and A_2 do not connect, then we know that the angle $A_1 0 A_2$ facing A_{2k} is either too small or too large. If it is too small, then we get that all $A_1 A_{2\ell+1}$ distances will be smaller than $A_1 A_2$ and hence A_1 will have degree 0. This is not possible since we know that A_1 must have at least 2 odd indexed neighbors. On the other hand, if the angle is too large, that implies that its complement in too small. In this case, any edge of the form $A_{2\ell_1} A_2$ will be smaller than $A_1 A_2$ and therefore A_2 would have degree 0. This would again cause a contradiction with the fact that they participate in a cycle.

In order to prove that A_2 must also connect to A_3, we first use the fact that A_2 must have another odd indexed neighbor A_i apart from A_1. If $A_i = A_3$, then we have our claim. If, on the other hand, $A_i \neq A_3$, we can no longer be sure that A_2 connects to A_3. But then, A_3 would be situated between A_1 and A_i and so, by Property 1 in the proof of Lemma 5.3, we have that $A_3 \in \widehat{A_1 A_i}_{A_2} \subseteq I(A_2)$ and therefore, A_2 and A_3 must also be connected.

We now continue that argument and show that A_4 must be connected to A_3 and A_5. Suppose A_3 is not connected to A_4. Then, by similar arguments as before, we can argue that all the other odd points to the left of A_3 can't connect to A_4 either (i.e. $A_5, \ldots, A_{2k-1}$). But that only leaves A_1 as A_4's neighbor, and we know that A_4 has degree at least two. Therefore, A_3 must be connected to A_4.

Now assume that A_4 is not connected to A_5. By the same reasoning as before,

no other higher indexed odd points can connect to A_4. in that case, we have that A_4 must connect to A_1 and A_3 in the cycle. In addition to this, the same holds for A_2: if there is a point A_i that covers A_2, then that point must also cover A_4. We are then left with the situation that A_2 and A_4 can only be connected to A_1 and A_3. However, in the original cycle (containing more than four points), A_2 and A_4 need to connect to at least 3 different odd vertices. If they only connect to two odd vertices, then we will have cycle on exactly four points which satisfies our initial constraints.

The previous argument can be made more general to show that $A_{2\ell}$ must connect to $A_{2\ell-1}$ and $A_{2\ell+1}$. Suppose, to start with, that $A_{2\ell-1}$ does not connect to $A_{2\ell}$. Then $A_{2\ell+1}, \ldots, A_{2k-1}$ cannot be connected to $A_{2\ell}$ either. Looking at $I(A_{2\ell+1})$, we know that it must be fully contained on either side of $A_{2\ell}$. Suppose it is on the right, i.e. in the arc $\widehat{A_{2\ell+2}A_{2k}}$. In other words, it cannot connect to any of the points $A_2, \ldots, A_{2\ell}$. We then also get that $A_{2\ell-1}, A_{2\ell+1}, \ldots A_{2k-1}$ cannot connect to $A_2, \ldots, A_{2\ell}$. Therefore, in the original cycle, the points $A_2, \ldots, A_{2\ell}$ must be covered solely by the remaining odd points $A_1, A_3, \ldots, A_{2\ell-3}$. In other words, we have that ℓ distinct even points can only be connected to $\ell - 1$ distinct odd points. But, as noticed before, in any subpath of a cycle, ℓ even points need at least $\ell + 1$ distinct odd neighbors (if they have exactly ℓ, then they must form a cycle).

Now suppose that $I(A_{2\ell-1})$ is fully contained in the left side of $A_{2\ell}$, i.e. $I(A_{2\ell-1}) \subseteq \widehat{A_2A_{2\ell-2}}$. Then notice that $A_{2\ell}$ can only connect to $A_2, A_4, \ldots, A_{2\ell-2}$. Furthermore, the same is true for $A_1, A_3, \ldots, A_{2\ell-3}$ as well. Overall we have that the points $A_1, A_3, \ldots, A_{2\ell-1}$ can only be covered by the points $A_2, A_4, \ldots, A_{2\ell-2}$.

In other words, we have ℓ distinct odd points that can only be connected to $\ell - 1$ distinct even points. We run into the same situation as before. We can therefore conclude that $A_{2\ell-1}$ must connect to $A_{2\ell}$.

We are left to show now that $A_{2\ell}$ must also connect to $A_{2\ell+1}$. Applying the same reasoning from before would give us that $A_2, A_4, \ldots, A_{2\ell}$ can only be connected to $A_1, A_3, \ldots, A_{2\ell-1}$. In other words, we would have ℓ even points that can only be connected to ℓ odd points. If the cycle has a total number of 2ℓ points, then we would get a cycle with the desired properties. Otherwise, the number of odd points would not be enough to be able to connect the ℓ even points to the rest of the cycle.

In a similar manner, we can look from the perspective of the odd vertices and argue that A_1 needs to be connected to A_2 and A_4, A_3 needs to be connected to A_4 and A_6 and, in general, $A_{2\ell-1}$ needs to be connected to $A_{2\ell}$ and $A_{2\ell+2}$. These edges, will, in turn, give us the path $A_2 - A_1 - A_4 - A_3 \ldots - A_{2k}$.

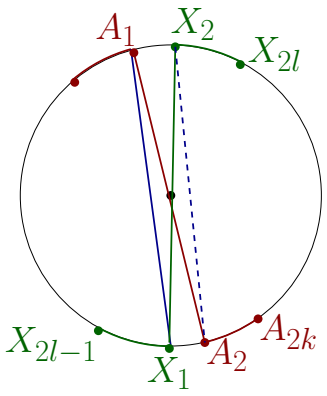
Moreover, by using all the edges we have shown must exist, we can also construct a cycle that visits all the points. The easiest way to construct is by inductively starting with the cycle $A_1 - A_2 - A_3 - A_4$. We can extend this cycle by replacing the edge $A_3 - A_4$ with the path $A_3 - A_6 - A_5 - A_4$, to get the cycle $A_1 - A_2 - A_3 - A_6 - A_5 - A_4$. The most important thing to notice about this cycle is that it contains the edges A_1A_2 and $A_{2k-1}A_{2k}$.

□

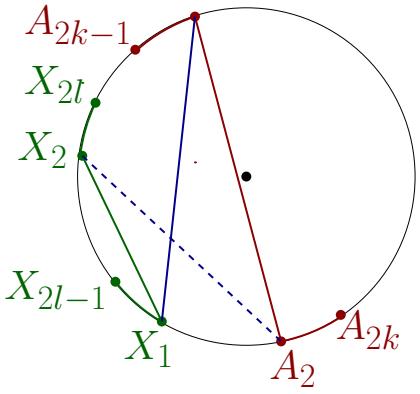
We have now shown that each even cycle can be ordered such that it can be transformed into any of these structures:

- a path between A_1 and A_{2k}

- a path between A_2 and A_{2k-1}
- a cycle that connects A_1 with A_2 and A_{2k-1} with A_{2k}.



(a) The cycles cross, i.e. their coverage arcs are on opposite sides with respect to the other cycle.



(b) The cycles do not cross.

Figure 5.7: The two cycles are drawn in red and green. The blue edge exists because the cycles are neighboring. Dashed blue edges do not exist.

Now we are in a good position to determine when two cycles can be merged. Consider the two cycles as ordered in Fig. 5.7. We assumed, without loss of generality, that the point X is on the arc $\overset{\frown}{A_2A_{2k-1}}$. From Lemma 5.4, we get that A will either be A_1 or A_{2k} (i.e. those are the two boundary points guaranteed to be searching that arc). We distinguish between two cases, based on where the other coverage arc of the second cycle is situated: does the second cycle cross the first one or not?

Case 1: the cycles cross (Fig. 5.7(a)). In other words, the coverage arcs of the second cycle are on opposite sides of the first cycle's coverage arcs (if the first cycle does not consist of two disjoint coverage arcs, then we are in Case 2). In this case, we can either have that A_1 connects to X_1 or that A_{2k} connects to X_{2l-1}. In

the first case, notice that if the edge A_2X_2 also exists, we can merge the two cycles into another cycle. Specifically, we can invoke Lemma 5.5 and get the first cycle connecting A_1 with A_2 and the second one connecting X_1 with X_2. We can then replace these edges with A_1X_1 and A_2X_2 and be done.

If, on the other hand, the edge A_2X_2 does not exist, then we claim that $A_2, \ldots, A_{2k}$ cannot connect to $X_2, \ldots, X_{2l}$. First notice that these points are contained on the same side of the line A_2X_2, opposite from the points A_1 and X_1. Let $\widehat{A_2X_2}$ be the arc that contains $A_2, \ldots, A_{2k}$ and $X_2, \ldots, X_{2l}$. Since the edge A_2X_2 does not exist, that means that the central angle of $\widehat{A_2X_2}$ is either too small or too big. If the angle is small, that means that all the central angles between points in $A_2, \ldots, A_{2k}$ and points in $X_2, \ldots, X_{2l}$ are also going to be small. Therefore, none of those edges can exist. If on the other hand, the angle is too big, then its complement is too small and A_1 and X_1 cannot connect, which is a contradiction.

In the case in which the edge $A_{2k}X_{2l-1}$ exists (instead of A_1X_1), we can merge the two cycles when additionally we also have that the edge $A_{2k-1}X_{2l}$ exists. Otherwise, through similar arguments as before, we can conclude that $A_1, \ldots, A_{2k-1}$ cannot connect to $X_2, \ldots, X_{2l}$.

Case 2: the cycles do not cross (Fig. 5.7(b)). In this case, we claim that they can always be merged into a cycle. Since the two points are neighboring, we can either have that A_1 connects to X_1 or that A_2k connects to X_{2l}. If A_1X_1 exists then by the same reasoning as before, we can arugue that the points A_2 and X_2 must be connected. In this case, notice that the line A_2X_2 has the entire cycle $A_1, A_2 \ldots$ on one arc and the points $X_1, \ldots, X_{2l-1}$ and X_2 on the other arc. If A_2 and X_2

are not connected, one of those arcs is too small and therefore, cannot contain any connected points, which is a contradiction. Therefore, A_2X_2 must be an edge. The case in which the edge $A_{2k}X_{2l}$) exists can be treated in a similar way.

As noticed in Lemma 5.4, the only other case to be considered for this arrangement is when X_{2l-1} actually connects to A_{2k} or A_{2k-1} connects to X_{2l} (these are the only other cases in which two cycles can be neighboring).

5.3.4 Diverging Even Cycles

We have therefore identified the conditions under which we can merge two neighboring cycles. In particular, we have noticed that if the two neighboring cycles do not cross each other, they can always be merged. We will from now on only focus on cycles that cross each other. Assume, throughout our argument, that our two neighboring cycles are ordered as in Fig. 5.7. We identify the cycle that we could not merge into cycles.

Definition 5.7. *The two cycles in Fig. 5.7(a) are said to be* ***diverging*** *if:*

1. *at least one of the edges A_1X_1 and A_2X_2 does not exist and*

2. *at least one of the edges $A_{2k-1}X_{2l}$ and A_{2k} and X_{2l-1} does not exist.*

As we have seen before, if two neighboring cycles are not diverging, then we can always merge them. We will show that otherwise, they cannot be merged.

Lemma 5.6. *Two diverging cycles can never be merged.*

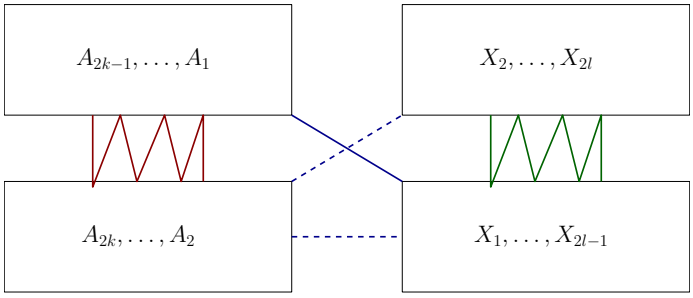


Figure 5.8: We represent the coverage arcs for each cycle. The red and green edges are part of the original cycles and link opposing coverage arcs. The blue edge represents the edge A_1X_1 and potentially other ones. The dashed edges mean that no edges between any points in those coverage arcs can exist.

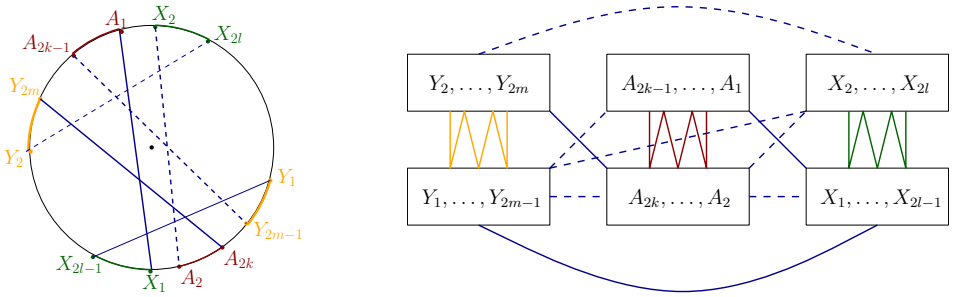
Proof. We will consider the case in which two out of the four edges exist. The other cases can be treated the same. Without loss of generality, we assume that the edges A_1X_1 and $A_{2k-1}X_{2\ell}$ exist. As noted before, the fact that the edge A_2X_2 does not exist implies that the arc $\overset{\frown}{A_2X_2}$ containing the points $A_2, \ldots, A_{2k}$ and $X_2, \ldots, X_{2\ell}$ is to small and therefore, none of the points on it can be connected. Similarly, we also get that the points $A_2, \ldots, A_{2k}$ and $X_1, \ldots, X_{2\ell-1}$ cannot be connected. In other words, we get the graph in Fig. 5.8. In particular, we have that the only possible neighbors of $A_{2k}, \ldots, A_2$ are the points $A_1, \ldots, A_{2k-1}$. Not only can those even points not connect to points in the second cycle but they also cannot connect with each other. In other words, we have k points who collectively can be covered by only k other points. But this means that they can never be part of a bigger cycle because then they would require at least $k+1$ distinct neighbors.

□

We can now ask the question of whether N pairwise diverging cycles can be merged. We consider the same scenario as before and show that, if we add a third

cycle that diverges with both of them, there is no way to merge them to a cycle.

In the two cycle case, we argued that there is no way to connect the points $A_2, \ldots, A_{2k}$ to a bigger cycle because they didn't have enough neighbors. Our strategy for the three cycle case is going to be the same: our goal will be to identify a coverage arc that does not have any neighbors beyond the ones in the original cycle it belongs to. Naturally, if $A_2, \ldots, A_{2k}$ continues to have no new neighbors in this third cycle, then we are done.



(a) Three diverging cycles.

(b) Corresponding coverage arcs for each cycle, together with edges between them representing whether they can connect or not.

Figure 5.9: Notice that, because of divergence, each blue edge that exists has a corresponding/opposing dashed blue edge that cannot exist.

Lemma 5.7. *Multiple pairwise diverging cycles can never be merged.*

Proof. Assume that the third cycle provides new neighbors for $A_2, \ldots, A_{2k}$. In other words, we assume that the cycle $Y_1, \ldots, Y_{2m}$ has a point that connects to some point in $A_2, \ldots, A_{2k}$. First, we notice that this new cycle must cross with both of the cycles, as in Fig. 5.9(a). This means that one coverage arc $Y_1, \ldots Y_{2m-1}$ is on the arc X_{2l} and A_{2k} and the other arc is in between A_{2k-1} and X_{2l-1}. From this, we immediately get that the points $X_2, \ldots, X_{2\ell}$, $A_2, \ldots, A_{2k}$ and $Y_1, \ldots, Y_{2m-1}$ cannot connect, since they are on the small arc A_2X_2 from before. Because of Lemma 5.4,

we get that if we want $A_2, \ldots, A_{2k}$ to connect to the third cycle, the only connection possible is between A_{2k} and Y_{2m}. This is because the only other option would be for A_2 to connect to Y_1 but we dismissed that observation from before. Now, since we have that A_{2k} connects to Y_{2m} and the A and Y cycles diverge, we also get that the edge $A_{2k-1}Y_{2m-1}$ cannot exist because it is too small. This now implies that the arc $\widehat{A_{2k-1}Y_{2m-1}}$ opposing the edge $A_{2k}Y_{2m}$ is too small and therefore, none of the points $A_1, \ldots, A_{2k-1}$, $X_2, \ldots, X_{2\ell}$ and $Y_1, \ldots, Y_{2m-1}$ can connect.

At this point, we have two coverage arcs that are possible candidates for our argument: $Y_1, \ldots, Y_{2m-1}$ and $X_2, \ldots, X_{2\ell}$. We have shown that $Y_1, \ldots, Y_{2m-1}$ cannot connect to any of the other coverage arcs except $X_1, \ldots, X_{2\ell-1}$. If we have that $Y_1, \ldots, Y_{2m-1}$ does not connect to $X_1, \ldots, X_{2\ell-1}$, then we are done. Otherwise, assume that they do connect and get that the edge $Y_1X_{2\ell-1}$ must exist, according to Lemma 5.4. But since the Y and the X cycles are divergent, we then get that the edge $Y_2X_{2\ell}$ cannot exist and the arc $\widehat{Y_2X_{2\ell}}$ opposing $Y_1X_{2\ell-1}$ is too small. We then get that none of the points $Y_2, \ldots, Y_{2m}$, $A_1, \ldots, A_{2k-1}$ and $X_2, \ldots, X_{2\ell}$ can connect. This in turn means that the coverage arc $X_2, \ldots, X_{2\ell}$ does not connect to any other points except the ones in its original cycle.

We can repeat this argument inductively and show that N pairwise diverging cycles cannot be merged into a cycle.

□

5.3.5 Polynomial Time Algorithm

We are now ready to describe a polynomial time algorithm for points on a circle. We will then discuss the specific properties that are needed in order for this algorithm to go through for more general cases. The pseudocode for the algorithm is given below.

We start by a running a maximum 2-matching algorithm on our input points [130]. As we have seen in Chapter 3, this can be done in time $\mathcal{O}(n^2 \log n(m+n \log n))$, where n is the number of vertices and m is the number of edges [131]. By definition, this generates a subgraph such that each vertex has degree at most 2. This in turn corresponds to a collection of edges and cycles. For our purposes, we consider edges to be cycles on two points. The time to compute the 2-matching dominates the overall runtime.

We do a preprocessing step where, for each even cycle, we detect the endpoints of their upper and lower coverage arcs, as we do in Lemma 5.5. We also keep track of which cycle each point belongs to and further, in the case of even cycles, whether they are in the upper or lower coverage arc. This can be done in $\mathcal{O}(n)$ time by going around the circle.

Out of the cycles we obtain, we distinguish between the case in which one odd cycle is present and not. If we encounter an odd cycle, we use Lemma 5.2 to argue that every other cycle computed must interleave with this odd cycle and then apply the strategy described in Lemma 5.3 to merge them with the odd one. We do one merge step per cycle and it takes $\mathcal{O}(n)$ per merge to detect where the witness for

interleaving happens and to relabel all the vertices to belong to the new cycle. In the worst case, this takes $\mathcal{O}(n^2)$ time overall.

If, on the other hand, there are no odd cycles present and we first consider pairwise interleaving and merge those cycles that do interleave. We can detect this by going around the circle and detecting when a point that belongs to one cycle appears among points that belong to another cycle's upper or lower coverage. Each time we encounter an occurrence, we can update the label of the points. Once we no longer encounter interleaving cycles, we can easily detect neighboring cycles by applying Lemma 5.4: it is enough to look at the points right next to the boundary of the cycle. By Lemma 5.6, we know that there must exist a neighboring cycle that we can merge with. Otherwise, we could never compute a Hamiltonian cycle and we return false. We continue this until all cycles have been merged. This case can also be performed in $\mathcal{O}(n^2)$.

5.4 The General Case

In this section, we discuss more general cases for which our approach applies.

5.4.1 Separability as a General Principle

We identify here the general principle that lies behind our proofs in Section 5.3. Our goal is to reduce our technique to a property that can be easily checked for a set of input points and from which our polynomial time algorithm follows easily. Specifically, we identify the principle as the input points having the property of

Algorithm 1 One Circle Case

Input: A set of points on a circle and a guess for D, the length of min edge in the Max Scatter TSP

Output: A tour if the guess is good, otherwise the guess is too small.

```
1: Construct the graph G given the threshold D
2: Compute a maximum 2-matching on G, obtain a series of cycles of G
3: if there are any odd cycles then
4:    take one odd cycle
5:    while there are even cycles do
6:       merge the odd cycle with the even one because they interleave
7:       update the odd cycle
8:    you have only odd cycles left, pick one cycle C
9:    while there are still other odd cycles do
10:      merge one odd cycle with C because they interleave
11:      update C to correspond to the new cycle
12: else
13:   while there are interleaving cycles do
14:      merge them
15:   compute boundary points for each even cycle
16:   while there are neighboring cycles do
17:      pick a pair that you can merge
18:      check that the new cycle merges with other existing cycles
19:      if the remaining cycles cannot be merge then
20:         exit, the guess is too small
```

separability:

Definition 5.8. *For a given guess scatter, we say that an input set of points $\mathcal{P}$ is* ***separable*** *if it is convex and for all points P_i and P_j that do not connect, we have that there can be no edges between the input points contained in one of the induced hyperplanes.*

In the circle case, this property holds because the edge P_iP_j not existing corresponds to the angle $\angle P_iOP_j$ being too small or alternatively, its complement too small. Furthermore, each hyperplane induces two complementary arcs on the boundary of the circle, one whose central angle would be too small. It then follows that

no points lying on that arc could form edges. We also mention that if a point set is separable, then it also follows that all reachability areas must be continuous with respect to a fixed clockwise or counter-clockwse ordering on the points.

We also note that separability implies the fact that every reachability area consists of consecutive points in the ordering:

Lemma 5.8. *Given an ordering on the points, if $\mathcal{P}$ is separable, then the reachability area of every point consists of consecutive points.*

Proof. Suppose there exists a point P_i that does not connect to P_j but connects to points P_k and P_ℓ that are on different sides of P_j in the ordering. From the definition of separability, it follows that one of the two hyperplanes defined by P_iP_j contains points that cannot connect. From convexity, we have that P_k and P_ℓ belong to different hyperplanes. Since P_i belongs to both, it would then follow that either the edge P_iP_k or P_iP_ℓ cannot exist. □

Our proofs in Section 5.3 contain alternate arguments that specifically highlight when this principle is applied. We then get that:

Lemma 5.9. *If a point set is separable for a given guess scatter, then it admits a Hamiltonian tour that can be computed in polynomial time.*

Proof. Our algorithm is the same as the one for the circle case explained in Section 5.3.5. Notice that our algorithm is independent of the fact that the points are on a circle and just depend on the fact that we have an available ordering on the points for which the point set is separable. The order follows naturally from the convexity. □

We note that not all convex point sets are separable, as can be seen from Fig. 5.10. In this example, there exists a Hamiltonian tour but for the given guess scatter, the points are not separable. We also note that in this example, P_2's reachability are consists of the points P_4 and P_6 that are not consecutive on the convex hull.

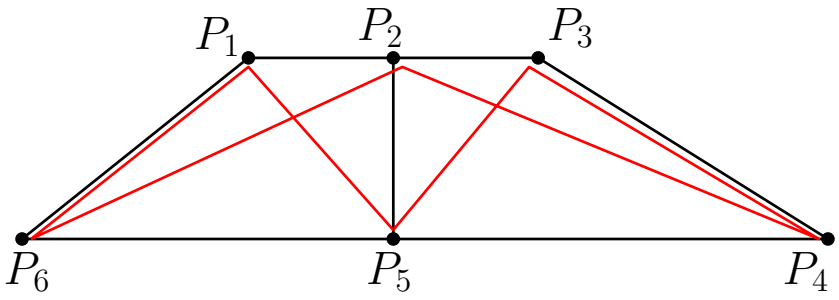


Figure 5.10: In this construction, $|P_2P_5| \leq |P_3P_4|$ and $|P_2P_5| \leq |P_1P_6|$. If we set the guess scatter at P_2P_5, we get that the point set is not separable. In red, we can see that a possible Hamiltonian tour.

We note that separability can be tested in $\mathcal{O}(n^2)$ by looking at the adjacency matrix of the associated threshold graph.

Chapter 6

Conclusion

In this thesis, we discussed several clustering and touring problems in the Euclidean plane. Our work was motivated by applications in robotics and logistics planning such as sensor placement and data collection. Our goal was to provide algorithms with provable theoretical guarantees that take advantage of the underlying geometry of the instances and improve on the state-of-the-art.

In Chapter 3, we formulated a new coverage constraint that we used to limit the uncertainty incurred by two sensors trying to localize a target. Existing techniques fall short in this new regime, as they are not well suited to deal with sensors that collaborate in order to cover a target. In this context, we designed a general framework that selects a small number of sensors and approximates the new coverage constraint to arbitrary precision in most cases. We applied our technique to obtain good approximation algorithms for a variety of sensor placements problems with additional line-of-sight or proximity constraints. In the course of our analysis, we provided additional approximations for specialized cases and also investigated

the computational complexity of the problem on general metric spaces. We mention here some specific open problems:

Problem 1. Extend the results of Lemma 3.1 to also apply for the case of $\alpha > \pi/3$. As of now, in order for the proof to work, it looks like the requirement that $\alpha \leq \pi/3$ is necessary in order to guarantee that the union of the two double-wedges corresponding to s_1 and s_2 is a larger double-wedge. Without that requirement, their union would not be a continous double-wedge and the optimal sensors might "fall between the cracks". However, we do not believe that we have exploited the full structure of the set S of already chosen sensors. In particular, the proof is concerned with adding sensors that $(\alpha - \epsilon)$-cover targets in T that are not yet already $(\alpha - \epsilon)$-covered by sensors in S. If a target t is not already $(\alpha - \epsilon)$-covered, however, it means that, from the perspective of t, all the sensors in S are either really close together (angle $< \alpha - \epsilon$) or really far apart (angle $> \pi - \alpha + \epsilon$), so in some sense, it might not matter what our choice for s_1 and s_2 is. In this way, we might still be able to pick s_1 and s_2 in a way that guarantees that the union of their respective double-wedges will be a continuous double-wedge that must necessarily contain one of the optimal sensors.

Problem 2. in Section 3.4, we proposed other bi-criteria approximations for the case of angular coverage and distance constraints. At the cost of relaxing the distance constraint by factors of 2 and 3, we were able to obtain exponential improvements in the approximation factor for the number of sensors selected. We achieved this by designing small hitting sets for geometric objects that include our original sets but are strictly larger. We also employed the shifting technique of Hochbaum

and Maass [128] to combine multiple local hitting sets into a global one whose size does not depend on the diameter of the input points. It would be interesting to see in what other situations these techniques apply. Conventionally, constructing small hitting sets for a certain type of objects is done by exploiting properties of those objects such as fatness, combinatorial complexity or VC dimension. When it come to the potential applications of hitting sets, however, it might be meaningful to consider relaxed versions in which the point chosen are guaranteed to be within a certain distance of each object chosen.

Problem 3. Extend the framework to incorporate other measures of uncertainty, such as the one introduced by Tekdas et al [120]. As we have seen, by setting $R = \sqrt{U \sin \alpha}$, we can guarantee that the uncertainty will be smaller than U. The choice for α, however, can be arbitrary. It is unclear if there exists a specific choice for α that would give us the smallest approximation factor. We could try to devise a result similar to that of Lemma 3.1, which would require that we take into consideration how small changes in α can affect the overall uncertainty. Alternatively, we could consider an additional probability distribution on the uncertainty measure, and give more priority to the areas in which we have low confidence, in the manner of Kraus et al [25]. In this sense, the problem would relate to a weighted version of α-Ang etc, in which we try to maximize the total weight of the targets covered.

Problem 4. Incorporate motion and compute a shortest path that visits the targets and takes measurements that respect the angle constraint. For this, we could first use our framework to compute the smallest number of sensing locations (the locations where we would previously place sensors) and then compute a TSPN tour

that visits all such locations. The analysis would have to compute the length of such a tour with the optimal length tour that could potentially resemble an Angular-Metric TSP tour that, when visiting each target, changes direction in a bounded way. In other words, it takes one measurement as it enters the target and then a second measurement as it leaves, but from a perspective that is well separated angularly. This would amount to computing an Angular-Metric TSP tour in which each direction change is bounded, which is one of the open problems identified by Aggarwal et al [46].

In Chapter 4, we investigated the Traveling Salesman Problem with Neighborhoods when the neighborhoods are uniform disks. This is an important generalization of the famous Traveling Salesman Problem and our particular motivation came from the task of gathering data from sensors that don't have a communication infrastructure sand need to be visited by a tour. A fundamental question that arises from this generalization compares the optimal TSP tour on the centers of the disks with the TSPN solution. In particular, what is the gain that we get from visiting the disks at their boundaries rather than at their center? We investigate this question and show nuanced results that either improve on known bounds or show better lower bounds on the length of the optimal TSPN tour. As a consequence, we show an improved constant factor approximation for TSPN. We mention some natural open problems:

Problem 5. The intuition behind our new lower bound is that an optimal TSPN tour should not visit a disk multiple times unless it is supported by a straight line. While this is true for TSP, it is no longer true for TSPN. In fact, there are

natural structures for which this can happen, which we identify as β-triads. In describing such instances, the main property of disks that we use is the fact that their tangents from a point can be determined exactly. We believe a similar property can be described for objects that have a large diameter or more specifically, are fat. The next step in our method would therefore to apply our technique for such classes of objects.

Problem 6. For the case of overlapping disks, we used the algorithm of Dumitrescu and Tóth [143] that first computes a tour on a maximal subset of disjoint disks and then augments that tour to visit the remaining disks. We incorporated our analysis by updating the lower bounds that we get on the tour on the disjoint disks and that lead to an improved overall factor, as expected. It would be interesting to see how our structural understanding behaves in this new setting as well. For instance, how does a β-triad behave in the presence of additional overlapping disks? On one hand, if there are no such overlapping disks, then we don't need to augment our tour locally. On the other hand, if there are overlapping disks, that presents an opportunity for us to choose how we visit them locally. Specifically, can we modify our initial maximal set such that we construct more β-triads and obtain better bounds?

Problem 7. For our proof of the Häme, Hyytiä and Hakula conjecture for $n = 3$ in Section 4.5, we used a transformation in which we mapped all the points at which the TSPN touches each disk to points on the boundary of a fixed circle. Using the theory of Fermat-Weber points, we then showed that the detour bound can be expressed as a Maximum TSP tour on those points. Understanding the structure of

instances in which this Maximum TSP tour has a large optimal value (i.e. close to $2Rn$) can help us describe the cases in which the TSPN tour on the disks is much smaller than the TSP tour on the centers. Furthermore, showing that the Maximum TSPN tour corresponds to a Maximum Scatter TSP tour in the circle case would allow us to understand the behaviour of the TSPN tour locally rather than just globally. Since the Maximum Scatter TSP case is well understood in the circle case, this could also provide us a full description of how both of these tours look like on the circle.

Problem 8. The method of considering detour bounds can be extended to allow for different cost functions. For example, in the Data Gathering framework of Tekdas et al [48], one needs to decide how close to get to the sensors in order to download the data. The interesting trade-off is that the closer we get to the sensor, the faster we can download the data. The final cost of the tour obtained consists not just of the length of the tour computed but also of the time it takes to download data from each sensor. Similarly, one can imagine a scenario in which we would have fixed radii but the cost of traveling inside and outside the disks might be different. In all of these scenarios, the algorithm of Dumitrescu and Mitchell [68] lends itself nicely because it uses the TSP on the centers as a common benchmark. It would be interesting to see how our framework behaves in front of these scenarios. Specifically, can we control the type of edges that we get by setting the radius profiles accordingly? How does the download cost feature into our detour bound and how does that carry through our analysis?

Finally, in Chapter 5, we visit the Maximum Scatter TSP in more detail.

Specifically, we are interested in a long standing open problem that asks whether the two dimensional Euclidean case is polynomially time solvable. The answer is known for points on a line, a circle and for some type of rectangular grids but the solutions are highly specialized and the geometry is used in indirect ways. We provide an alternate solution for the case of points on a circle and determine a specific property that the input points must have in order for our algorithm to apply. We then further refine our algorithm and discuss more cases for which we believe it applies. We mention some directions in which we plan to follow our work:

Problem 9**.** The property of separability that we have identified as underlying our algorithm refers to a unit disk graph (i.e. threshold graph) on points in a convex position. We can extend the property to points in general position for which an ordering exists that exhibits separability. It would be interesting to further investigate this line of thought and identify if known graph classes exhibit it naturally. This would give us a more systematic understanding of the cases for which polynomial time algorithms exist and furthermore, point us to potential construction for showing NP-hardness.

Problem 10**.** In line with Problem 7, understanding whether Max Scatter TSP is the same as Max TSP for points on a circle could point to a more general equivalence. There are convex instances for which we know that they cannot be equivalent, but it would be worthwhile to understand the scope of it. The idea would be to establish a framework that links our algorithm for Max Scatter on the circle to one for Max TSP and therefore generalize our results to the latter. Specifically, we can ask whether Max TSP is easy to solve on separable instances.

Bibliography

[1] Janardhan Kulkarni and Sathish Govindarajan. New ϵ-net constructions. In *Proceedings of the 22nd Annual Canadian Conference on Computational Geometry, Winnipeg, Manitoba, Canada, August 9-11, 2010*, pages 159–162, 2010.

[2] Deepak Bhadauria, Volkan Isler, Andrew Studenski, and Pratap Tokekar. A robotic sensor network for monitoring carp in minnesota lakes. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3837–3842. IEEE, 2010.

[3] PG Bajer, CJ Chizinski, and PW Sorensen. Using the judas technique to locate and remove wintertime aggregations of invasive common carp. *Fisheries Management and Ecology*, 18(6):497–505, 2011.

[4] Dan Taylor and Larry Katahira. Radio telemetry as an aid in eradicating remnant feral goats. *Wildlife Society Bulletin (1973-2006)*, 16(3):297–299, 1988.

[5] Pratap Tokekar, Elliot Branson, Joshua Vander Hook, and Volkan Isler. Tracking aquatic invaders: Autonomous robots for monitoring invasive fish. *Robotics & Automation Magazine, IEEE*, 20(3):33–41, 2013.

[6] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[7] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[8] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pages 5029–5037, 2017.

[9] Clemens Rösner and Melanie Schmidt. Privacy preserving clustering with constraints. *arXiv preprint arXiv:1802.02497*, 2018.

[10] Joseph O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, Inc., New York, NY, USA, 1987.

[11] William Scott, Gerhard Roth, and Jean-François Rivest. View planning for automated 3d object reconstruction inspection. *ACM Computing Surveys*, 35(1), 2003.

[12] Kaikai Sheng, Zhicheng Gu, Xueyu Mao, Xiaohua Tian, Weijie Wu, Xiaoying Gan, and Xinbing Wang. The collocation of measurement points in large open indoor environment. In *Proceedings of the IEEE INFOCOM*, 2015.

[13] Fu-bao Wang, Long Shi, and Feng-yuan Ren. Self-localization systems and algorithms for wireless sensor networks. *Ruan Jian Xue Bao(J. Softw.)*, 16(5):857–868, 2005.

[14] Cha Zhang, Dinei Florêncio, Demba E Ba, and Zhengyou Zhang. Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings. *Multimedia, IEEE Transactions on*, 10(3):538–548, 2008.

[15] Ankur Kamthe, Lun Jiang, Matthew Dudys, and Alberto Cerpa. Scopes: Smart cameras object position estimation system. In *Wireless Sensor Networks*, pages 279–295. Springer, 2009.

[16] J.E.D. Williams. *From sails to satellites: the origin and development of navigational science*. Oxford University Press, 1992.

[17] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 2, pages 472–476 vol.2, 2001.

[18] Zoë Abrams, Ashish Goel, and Serge A. Plotkin. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proceedings of the Third International Symposium on Information Processing in Sensor Networks, IPSN 2004, Berkeley, California, USA, April 26-27, 2004*, pages 424–432, 2004.

[19] Samir Khuller, Robert Pless, and Yoram J. Sussmann. Fault tolerant k-center problems. *Theoretical Computer Science*, 242(12):237 – 245, 2000.

[20] Amol Deshpande, Samir Khuller, Azarakhsh Malekian, and Mohammed Toossi. Energy efficient monitoring in sensor networks. *Algorithmica*, 59(1):94–114, 2011.

[21] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Fault-tolerant clustering in ad hoc and sensor networks. In *26th IEEE International Conference on Distributed Computing Systems (ICDCS 2006), 4-7 July 2006, Lisboa, Portugal*, page 68, 2006.

[22] David Eppstein, Michael T. Goodrich, and Nodari Sitchinava. Guard placement for efficient point-in-polygon proofs. In *Proceedings of the 23rd ACM Symposium on Computational Geometry, Gyeongju, South Korea, June 6-8, 2007*, pages 27–36, 2007.

[23] Jocelyn Smith and William S. Evans. Triangle guarding. In *Proceedings of the 15th Canadian Conference on Computational Geometry, CCCG'03, Halifax, Canada, August 11-13, 2003*, pages 76–80, 2003.

[24] Noel A. C. Cressie. *Statistics for spatial data*. Wiley series in probability and mathematical statistics. J. Wiley & Sons, New York, Chichester, Toronto, 1993.

[25] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, June 2008.

[26] Randolph L Moses, Dushyanth Krishnamurthy, and Robert M Patterson. A self-localization method for wireless sensor networks. *EURASIP Journal on Applied Signal Processing*, pages 348–358, 2003.

[27] Neal Patwari, Joshua N Ash, Spyros Kyperountas, Alfred O Hero III, Randolph L Moses, and Neiyer S Correal. Locating the nodes: cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22(4):54–69, 2005.

[28] Andreas Savvides, Wendy Garber, Sachin Adlakha, Randolph Moses, and Mani B Srivastava. On the error characteristics of multihop node localization in ad-hoc sensor networks. In *Information Processing in Sensor Networks*, pages 317–332. Springer, 2003.

[29] A. Savvides, W.L. Garber, R.L. Moses, and M.B. Srivastava. An analysis of error inducing parameters in multihop sensor node localization. *Mobile Computing, IEEE Transactions on*, 4(6):567–577, Nov 2005.

[30] Tobias Mömke and Ola Svensson. Approximating graphic tsp by matchings. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 560–569. IEEE, 2011.

[31] Marcin Mucha. 13/9 -approximation for graphic tsp. *Theory of computing systems*, 55(4):640–657, 2014.

[32] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.

[33] Jiawei Qian, Frans Schalekamp, David P Williamson, and Anke Van Zuylen. On the integrality gap of the subtour lp for the 1, 2-tsp. *Mathematical Programming*, 150(1):131–151, 2015.

[34] José Correa, Omar Larré, and José A Soto. Tsp tours in cubic graphs: beyond 4/3. *SIAM Journal on Discrete Mathematics*, 29(2):915–939, 2015.

[35] Nishita Agarwal, Naveen Garg, and Swati Gupta. A 4/3-approximation for tsp on cubic 3-edge-connected graphs. *Operations Research Letters*, 46(4):393–396, 2018.

[36] Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998.

[37] Satish Rao and Warren D. Smith. Approximating geometrical graphs via "spanners" and "banyans". In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 540–550, 1998.

[38] Joseph S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric tsp, k-mst, and related problems. *SIAM J. Comput.*, 28(4):1298–1309, 1999.

[39] Ola Svensson, Jakub Tarnawski, and László A. Végh. A constant-factor approximation algorithm for the asymmetric traveling salesman problem. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 204–213, 2018.

[40] Suresh Nanda Kumar and Ramasamy Panneerselvam. A survey on the vehicle routing problem and its variants. 2012.

[41] Greg N Frederickson and Barry Wittman. Approximation algorithms for the traveling repairman and speeding deliveryman problems with unit-time windows. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 119–133. Springer, 2007.

[42] René Sitters. Polynomial time approximation schemes for the traveling repairman and other minimum latency problems. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 604–616. SIAM, 2014.

[43] Ke Chen and Sariel Har-Peled. The euclidean orienteering problem revisited. *SIAM Journal on Computing*, 38(1):385–397, 2008.

[44] Michael Khachay and Roman Dubinin. Ptas for the euclidean capacitated vehicle routing problem in $\mathcal{R}^d$. In *International Conference on Discrete Optimization and Operations Research*, pages 193–205. Springer, 2016.

[45] S'andor P. Fekete and Gerhard J. Woeginger. Angle-restricted tours in the plane. *Computational Geometry*, 8(4):195 – 218, 1997.

[46] Alok Aggarwal, Sanjeev Khanna, Rajeev Motwani, and Baruch Schieber. The angular-metric traveling salesman problem. *In Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, 29:221–229, 1997.

[47] Isabella Stock. *The Maximum Scatter TSP on a Regular Grid: How to Avoid Heat Peaks in Additive Manufacturing.* PhD thesis, Universität Bayreuth, 2017.

[48] O. Tekdas, V. Isler, Jong Hyun Lim, and A. Terzis. Using mobile robots to harvest data from sensor fields. *Wireless Communications, IEEE*, 16(1):22–28, February 2009.

[49] Bo Yuan, Maria Orlowska, and Shazia Sadiq. On the optimal robot routing problem in wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(9):1252–1261, 2007.

[50] Ming Ma, Yuanyuan Yang, and Miao Zhao. Tour planning for mobile data-gathering mechanisms in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 62(4):1472–1483, 2013.

[51] Gui Citovsky, Jie Gao, Joseph SB Mitchell, and Jiemin Zeng. Exact and approximation algorithms for data mule scheduling in a sensor network. In *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pages 57–70. Springer, 2015.

[52] Ben Popper. Ups researching delivery drones that could compete with amazons prime air. *The Verge, http://www. theverge. com/2013/12/3/5169878/ups-is-researchingits-own-delivery-drones-to-compete-withamazons.(Accessed 16.11. 2016)*, 2013.

[53] Niels Agatz, Paul Bouman, and Marie Schmidt. Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 2018.

[54] Stefan Poikonen, Xingyin Wang, and Bruce Golden. The vehicle routing problem with drones: Extended models and connections. *Networks*, 70(1):34–43, 2017.

[55] Xingyin Wang, Stefan Poikonen, and Bruce Golden. The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, 11(4):679–697, 2017.

[56] Robert Shuttleworth, Bruce L Golden, Susan Smith, and Edward Wasil. Advances in meter reading: Heuristic solution of the close enough traveling salesman problem over a street network. In *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 487–501. Springer, 2008.

[57] Deepak Bhadauria, Onur Tekdas, and Volkan Isler. Robotic data mules for collecting data over sparse sensor fields. *Journal of Field Robotics*, 28(3):388–404, 2011.

[58] Onur Tekdas, Deepak Bhadauria, and Volkan Isler. Efficient data collection from wireless nodes under the two-ring communication model. *The International Journal of Robotics Research*, 31(6):774–784, 2012.

[59] Petr Slavik. Errand scheduling problem. 1997.

[60] Francesco Carrabs, Carmine Cerrone, Raffaele Cerulli, and Manlio Gaudioso. A novel discretization scheme for the close enough traveling salesman problem. *Computers & Operations Research*, 78:163–171, 2017.

[61] Damon J Gulczynski, Jeffrey W Heath, and Carter C Price. The close enough traveling salesman problem: A discussion of several heuristics. In *Perspectives in Operations Research*, pages 271–283. Springer, 2006.

[62] William Kenneth Mennell. *Heuristics for solving three routing problems: Close-enough traveling salesman problem, close-enough vehicle routing problem, sequence-dependent team orienteering problem*. PhD thesis, 2009.

[63] Jing Dong, Ning Yang, and Ming Chen. Heuristic approaches for a tsp variant: The automatic meter reading shortest tour problem. In *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*, pages 145–163. Springer, 2007.

[64] Wei-Lun Chang, Deze Zeng, Rung-Ching Chen, and Song Guo. An artificial bee colony algorithm for data collection path planning in sparse wireless sensor networks. *International Journal of Machine Learning and Cybernetics*, 6(3):375–383, 2015.

[65] Jing-Sin Liu, Shao-You Wu, and Ko-Ming Chiu. Path planning of a data mule in wireless sensor network using an improved implementation of clustering-based genetic algorithm. In *Computational Intelligence in Control and Automation (CICA), 2013 IEEE Symposium on*, pages 30–37. IEEE, 2013.

[66] Alonzo Kelly. Precision dilution in triangulation based mobile robot position estimation. In *Intelligent Autonomous Systems*, volume 8, pages 1046–1053, 2003.

[67] Ioana O. Bercea, Volkan Isler, and Samir Khuller. Minimizing uncertainty through sensor placement with angle constraints. In *Proceedings of the 28th Canadian Conference on Computational Geometry, CCCG 2016, August 3-5, 2016, Simon Fraser University, Vancouver, British Columbia, Canada*, pages 287–294, 2016.

[68] Adrian Dumitrescu and Joseph SB Mitchell. Approximation algorithms for TSP with neighborhoods in the plane. *Journal of Algorithms*, 48(1):135–159, 2003.

[69] Adrian Dumitrescu and Csaba D Tóth. Constant-factor approximation for TSP with disks. In *A Journey Through Discrete Mathematics*, pages 375–390. Springer, 2017.

[70] Esther M Arkin and Refael Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Applied Mathematics*, 55(3):197–218, 1994.

[71] Mark de Berg, Joachim Gudmundsson, Matthew J Katz, Christos Levcopoulos, Mark H Overmars, and A Frank van der Stappen. Tsp with neighborhoods of varying size. *Journal of Algorithms*, 57(1):22–36, 2005.

[72] Khaled Elbassioni, Aleksei V Fishkin, and René Sitters. Approximation algorithms for the Euclidean traveling salesman problem with discrete and continuous neighborhoods. *International Journal of Computational Geometry & Applications*, 19(02):173–193, 2009.

[73] Lauri Häme, Esa Hyytiä, and Harri Hakula. The Traveling Salesman Problem with Differential Neighborhoods. In *European Workshop on Computational Geometry (EuroCG)*, Morschach, Switzerland, Mar. 2011.

[74] Christian L Müller. Finding maximizing Euclidean TSP tours for the Häme-Hyytiä-Hakula conjecture. Technical report, Technical Report CGL-TR-13, ETHZ, 2011.

[75] Hideki Ando, Yoshinobu Oasa, Ichiro Suzuki, and Masafumi Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5):818–828, 1999.

[76] Stephen L Smith, Mireille E Broucke, and Bruce A Francis. Curve shortening and the rendezvous problem for mobile autonomous robots. *IEEE Transactions on Automatic Control*, 52(6):1154–1159, 2007.

[77] Kai-Seng Chou and Xi-Ping Zhu. *The curve shortening problem*. CRC Press, 2001.

[78] Alexander Barvinok, Sándor P Fekete, David S Johnson, Arie Tamir, Gerhard J Woeginger, and Russ Woodroofe. The geometric maximum traveling salesman problem. *Journal of the ACM (JACM)*, 50(5):641–664, 2003.

[79] Sándor P Fekete. Simplicity and hardness of the maximum traveling salesman problem under geometric distances. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 337–345. Society for Industrial and Applied Mathematics, 1999.

[80] Esther M Arkin, Yi-Jen Chiang, Joseph SB Mitchell, Steven Skiena, and Tae-Cheon Yang. On the maximum scatter tsp. In *SODA*, volume 97, pages 211–220. Citeseer, 1997.

[81] Isabella Hoffmann, Sascha Kurz, and Jörg Rambau. The maximum scatter tsp on a regular grid. In *Operations Research Proceedings 2015*, pages 63–70. Springer, 2017.

[82] László Kozma and Tobias Mömke. Maximum scatter tsp in doubling metrics. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 143–153. SIAM, 2017.

[83] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[84] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[85] Avi Wigderson. Knowledge, creativity, and p versus np, 2006.

[86] David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256 – 278, 1974.

[87] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Math.*, 13(4):383–390, January 1975.

[88] SK Stein. Two combinatorial covering theorems. *Journal of Combinatorial Theory, Series A*, 16(3):391–397, 1974.

[89] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[90] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM (JACM)*, 41(5):960–981, 1994.

[91] Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 475–484. ACM, 1997.

[92] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[93] Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *J. ACM*, 33(3):533–550, May 1986.

[94] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985.

[95] Wen-Lian Hsu and George L Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979.

[96] Rina Panigrahy and Sundar Vishwanathan. An o(log* n) approximation algorithm for the asymmetric p-center problem. *J. Algorithms*, 27(2):259–268, 1998.

[97] Aaron Archer. Two O ($\log^*$ k)-approximation algorithms for the asymmetric k-center problem. In *Integer Programming and Combinatorial Optimization, 8th International IPCO Conference, Utrecht, The Netherlands, June 13-15, 2001, Proceedings*, pages 1–14, 2001.

[98] Inge Li Gørtz and Anthony Wirth. Asymmetry in k-center variants. *Theor. Comput. Sci.*, 361(2-3):188–199, 2006.

[99] Deeparnab Chakrabarty, Prachi Goyal, and Ravishankar Krishnaswamy. The non-uniform k-center problem. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 67:1–67:15, 2016.

[100] Judit Bar-Ilan, Guy Kortsarz, and David Peleg. How to allocate network centers. 15:385–415, 11 1993.

[101] Samir Khuller and Yoram J Sussmann. The capacitated k-center problem. *SIAM Journal on Discrete Mathematics*, 13(3):403–418, 2000.

[102] Marek Cygan, MohammadTaghi Hajiaghayi, and Samir Khuller. Lp rounding for k-centers with non-uniform hard capacities. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 273–282. IEEE, 2012.

[103] Gagan Aggarwal, Rina Panigrahy, Tomás Feder, Dilys Thomas, Krishnaram Kenthapadi, Samir Khuller, and An Zhu. Achieving anonymity via clustering. *ACM Transactions on Algorithms (TALG)*, 6(3):49, 2010.

[104] Sara Ahmadian and Chaitanya Swamy. Approximation algorithms for clustering problems with lower bounds and outliers. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 69:1–69:15, 2016.

[105] Moses Charikar, Samir Khuller, David M. Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.*, pages 642–651, 2001.

[106] Marek Cygan and Tomasz Kociumaka. Constant factor approximation for capacitated k-center with outliers. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 25. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[107] Dorit S Hochbaum and Wolfgang Maass. Fast approximation algorithms for a nonconvex covering problem. *Journal of algorithms*, 8(3):305–323, 1987.

[108] David Haussler and Emo Welzl. Epsilon-nets and simplex range queries. In *Symposium on Computational Geometry*, pages 61–71, 1986.

[109] H. Brönnimann and M.T. Goodrich. Almost optimal set covers in finite VC-dimension. *Discrete & Computational Geometry*, 14(1):463–479, 1995.

[110] Guy Even, Dror Rawitz, and Shimon (Moni) Shahar. Hitting sets when the vc-dimension is small. *Inf. Process. Lett.*, 95(2):358–362, July 2005.

[111] Pankaj K. Agarwal, Esther Ezra, and Micha Sharir. Near-linear approximation algorithms for geometric hitting sets. *Algorithmica*, 63(1-2):1–25, 2012.

[112] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.

[113] János Komlós, János Pach, and Gerhard Woeginger. Almost tight bounds for ϵ-nets. *Discrete Comput. Geom.*, 7(2):163–173, March 1992.

[114] Sören Laue. Geometric set cover and hitting sets for polytopes in $\mathbb{R}^2$. In *STACS 2008, 25th Annual Symposium on Theoretical Aspects of Computer Science, Bordeaux, France, February 21-23, 2008, Proceedings*, pages 479–490, 2008.

[115] Evangelia Pyrga and Saurabh Ray. New existence proofs ϵ-nets. In *Proceedings of the 24th ACM Symposium on Computational Geometry, College Park, MD, USA, June 9-11, 2008*, pages 199–207, 2008.

[116] Jirí Matousek. Reporting points in halfspaces. *Comput. Geom.*, 2:169–186, 1992.

[117] Tomás Feder and Daniel Greene. Optimal algorithms for approximate clustering. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 434–444, New York, NY, USA, 1988. ACM.

[118] Viswanath Nagarajan, Baruch Schieber, and Hadas Shachnai. The Euclidean k-supplier problem. In Michel Goemans and Jos Correa, editors, *Integer Programming and Combinatorial Optimization*, volume 7801 of *Lecture Notes in Computer Science*, pages 290–301. Springer Berlin Heidelberg, 2013.

[119] A. Efrat, S. Har-Peled, and J.S.B. Mitchell. Approximation algorithms for two optimal location problems in sensor networks. In *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*, pages 714–723 Vol. 1, Oct 2005.

[120] O. Tekdas and V. Isler. Sensor placement for triangulation based localization. *IEEE Tran. Automation Science and Engineering*, 7(3):681–685, 2010.

[121] V. Isler, S. Khanna, J. Spletzer, and C.J. Taylor. Target tracking with distributed sensors: The focus of attention problem. *Computer Vision and Image Understanding Journal*, (1-2):225–247, October-November 2005. Special Issue on Attention and Performance in Computer Vision.

[122] Guy Kortsarz. On the hardness of approximating spanners. *Algorithmica*, 30(3):432–450, 2001.

[123] Jiri Matousek. *Lectures on Discrete Geometry*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[124] Pavel Valtr. Guarding galleries where no point sees a small area. *Israel Journal of Mathematics*, 104(1):1–16, 1998.

[125] GautamK. Das, Robert Fraser, Alejandro Lpez-Ortiz, and BradfordG. Nickerson. On the discrete unit disk cover problem. In Naoki Katoh and Amit Kumar, editors, *WALCOM: Algorithms and Computation*, volume 6552 of *Lecture Notes in Computer Science*, pages 146–157. Springer Berlin Heidelberg, 2011.

[126] Alon Efrat and Sariel Har-Peled. Guarding galleries and terrains. In *Foundations of Information Technology in the Era of Networking and Mobile Computing, IFIP 17th World Computer Congress - TC1 Stream / 2nd IFIP International Conference on Theoretical Computer Science (TCS 2002), August 25-30, 2002, Montréal, Québec, Canada*, pages 181–192, 2002.

[127] Boris Aronov, Esther Ezra, and Micha Sharir. Small-size ϵ-Nets for Axis-Parallel Rectangles and Boxes. *SIAM J. Comput.*, 39(7):3248–3282, 2010.

[128] Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985.

[129] Judit Bar-Ilan, Guy Kortsarz, and David Peleg. How to allocate network centers. *J. Algorithms*, 15(3):385–415, 1993.

[130] Alexander Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. A*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003. Paths, flows, matchings, Chapters 1–38.

[131] Richard P. Anstee. A polynomial algorithm for b-matchings: An alternative approach. *Information Processing Letters*, 24(3):153 – 157, 1987.

[132] Sanjeev Arora, László Babai, Jacques Stern, and Z Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pages 724–733. IEEE, 1993.

[133] Dorit S Hochba. Approximation algorithms for np-hard problems. *ACM SIGACT News*, 28(2):40–52, 1997.

[134] Moses Charikar, MohammadTaghi Hajiaghayi, and Howard Karloff. Improved approximation algorithms for label cover problems. *Algorithmica*, 61(1):190–206, 2011.

[135] Leslie G. Valiant. Universality considerations in vlsi circuits. *Computers, IEEE Transactions on*, 100(2):135–140, 1981.

[136] Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5):753–782, 1998.

[137] Joseph SB Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM Journal on computing*, 28(4):1298–1309, 1999.

[138] Shmuel Safra and Oded Schwartz. On the complexity of approximating TSP with neighborhoods and related problems. *Computational Complexity*, 14(4):281–307, 2006.

[139] Cristian S Mata and Joseph SB Mitchell. Approximation algorithms for geometric tour and network design problems. In *Proceedings of the eleventh annual symposium on Computational geometry*, pages 360–369. ACM, 1995.

[140] Joachim Gudmundsson and Christos Levcopoulos. A fast approximation algorithm for TSP with neighborhoods. *Nord. J. Comput.*, 6(4):469, 1999.

[141] Joseph SB Mitchell. A constant-factor approximation algorithm for TSP with pairwise-disjoint connected neighborhoods in the plane. In *Proceedings of the twenty-sixth annual symposium on Computational geometry*, pages 183–191. ACM, 2010.

[142] T-H Hubert Chan and Khaled Elbassioni. A QPTAS for TSP with fat weakly disjoint neighborhoods in doubling metrics. *Discrete & Computational Geometry*, 46(4):704–723, 2011.

[143] Adrian Dumitrescu and Csaba D Tóth. The traveling salesman problem for lines, balls, and planes. *ACM Transactions on Algorithms (TALG)*, 12(3):43, 2016.

[144] Hans L Bodlaender, Corinne Feremans, Alexander Grigoriev, Eelko Penninkx, René Sitters, and Thomas Wolle. On the minimum corridor connection problem and other generalized geometric problems. *Computational Geometry*, 42(9):939–951, 2009.

[145] Joseph SB Mitchell. A PTAS for TSP with neighborhoods among fat regions in the plane. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 11–18. Society for Industrial and Applied Mathematics, 2007.

[146] T-H Hubert Chan and Shaofeng H-C Jiang. Reducing curse of dimensionality: Improved PTAS for TSP (with neighborhoods) in doubling metrics. *ACM Transactions on Algorithms (TALG)*, 14(1):9, 2018.

[147] Jacob E. Goodman, Joseph O'Rourke, and Csaba D Tóth, editors. *Handbook of Discrete and Computational Geometry, Third Edition*. CRC Press, 2017.

[148] Binay Bhattacharya, Jurek Czyzowicz, Peter Egyed, Godfried Toussaint, Ivan Stojmenovic, and Jorge Urrutia. Computing shortest transversals of sets. *International Journal of Computational Geometry & Applications*, 2(04):417–435, 1992.

[149] George O Wesolowsky. The Weber problem: History and perspectives. *Computers & Operations Research*, 1993.

[150] Alfred Weber. *Ueber den standort der industrien*, volume 2. 1909.