Interconnections in Theoretical Computer Science: A Survey of SETH, 3SUM, APSP, and Related Concepts

www.surveyx.cn

Abstract

The interconnected concepts of SETH, 3SUM, APSP, and related computational problems are pivotal in advancing our understanding of computational complexity, particularly in establishing conditional lower bounds and exploring the limits of efficient computation. This survey elucidates the intricate relationships between these concepts, highlighting their collective impact on theoretical computer science. The exploration of fine-grained reductions and dynamic lower bounds reveals significant insights into the adaptability of algorithms in dynamic environments, emphasizing the importance of energy efficiency in computational models. Recent advancements in universally optimal algorithms and dynamic k-mismatch algorithms demonstrate substantial improvements over existing methods, optimizing performance in real-world applications. The study of subquadratic algorithms, particularly in relation to NP-hardness, underscores the challenges in achieving efficient solutions for inherently difficult problems. The survey further explores the potential of quantum algorithms in addressing computational challenges, suggesting promising avenues for future research. By synthesizing these insights, the survey provides a comprehensive framework for understanding the complexities of various computational problems, driving innovation in algorithm design and optimization. Future research should focus on optimizing algorithms for dynamic and weighted graphs, exploring the implications of fine-grained complexity, and investigating the potential of quantum computing in computational geometry and related fields.

1 Introduction

1.1 Importance of Interconnected Concepts

Interconnected concepts in theoretical computer science, such as SETH, 3SUM, and APSP, are pivotal for advancing our understanding of computational complexity and efficient computation limits. These concepts form a network that collectively establishes lower bounds for various computational problems. The 3SUM conjecture, for instance, serves as a foundational basis for conditional lower bounds across numerous problems [1]. Recognizing these interconnections is essential for modeling complex systems, as emphasized in frameworks discussed by [2].

The development of universally optimal algorithms, as explored in [3] and [4], further illustrates the practical implications of these interconnected concepts. Such advancements are crucial in real-world applications, particularly within distributed networks where efficient optimization is necessary. The choice of appropriate distance closures in complex network analysis exemplifies how these interconnections can enhance theoretical understanding [5].

Moreover, the demand for breakthroughs in optimization algorithms, as discussed in [6], underscores the dynamic nature of these interconnected concepts. Existing algorithms often face challenges in adapting to real-time conditions, highlighting the need for deeper insights into the underlying

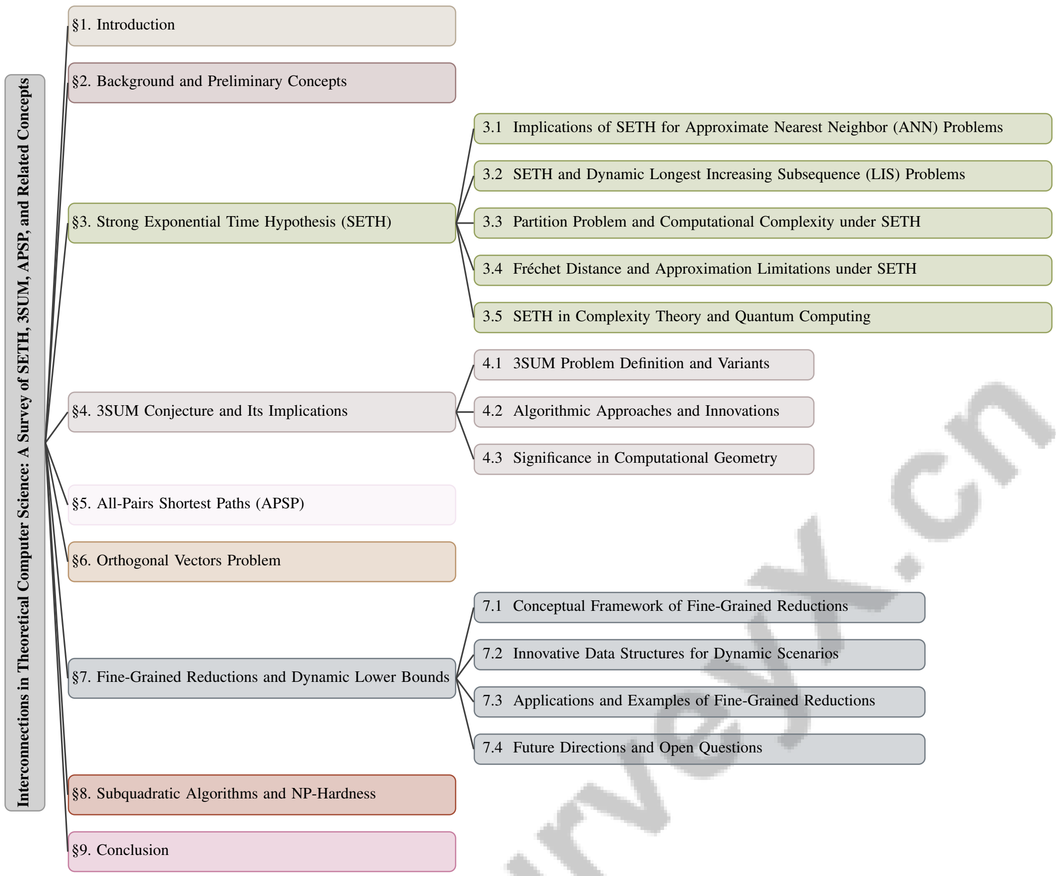

Figure 1: chapter structure

complexities. Thus, exploring these interconnections is vital for expanding the boundaries of computational feasibility and innovating solutions to complex theoretical problems.

1.2 Structure of the Survey

This survey is structured into several key sections, each addressing critical aspects of interconnected concepts in theoretical computer science. The introduction emphasizes the importance of understanding relationships among concepts like SETH, 3SUM, and APSP, and their collective impact on computational complexity.

Subsequent sections provide background and foundational concepts, setting the stage for an in-depth exploration of the theoretical frameworks driving research in computational complexity. The survey then analyzes individual concepts, beginning with the Strong Exponential Time Hypothesis (SETH), detailing its implications for SAT problems and its role in establishing computational lower bounds. This includes its relevance to Approximate Nearest Neighbor (ANN) and dynamic Longest Increasing Subsequence (LIS) problems, as well as its significance in quantum computing.

The examination of the 3SUM conjecture follows, highlighting its importance in computational geometry and its implications for complexity theory. The discussion of established reductions and conditional lower bounds underscores the conjecture's influence on various computational problems, such as dynamic data structures, graph algorithms, and pattern matching. Notable advancements have yielded polynomially higher lower bounds for critical problems, including optimal algorithms for enumerating triangles in graphs and maintaining dynamic longest increasing subsequences. Furthermore, new techniques have been introduced to demonstrate conditional lower bounds for dynamic versions of Maximum Cardinality Matching and specific string problems, reinforcing the conjecture's pivotal role in the complexity landscape [7, 8, 9, 1, 10].

AI-generated, for reference only.

The section on All-Pairs Shortest Paths (APSP) investigates its computational complexity, known algorithms, and connections to SETH and 3SUM, while addressing recent advancements and open questions in the field. Following this, the Orthogonal Vectors Problem is discussed, focusing on its relevance to fine-grained complexity and connections to other key problems, alongside algorithmic approaches and potential applications.

Fine-grained reductions and dynamic lower bounds are examined to elucidate their role in understanding efficient computation limits, discussing innovative data structures and suggesting future research directions. The survey concludes with an exploration of subquadratic algorithms and NP-hardness, addressing challenges posed by NP-hard problems and the significance of finding efficient algorithms, while considering dynamic and fine-grained complexity in pursuit of subquadratic solutions.

The synthesis of key insights emphasizes the significance of the interconnected concepts explored, particularly in network science and algorithm design. It underscores how the choice of distance closures in complex networks can substantially influence the interpretation of indirect associations and network structures. The discussion suggests promising future research avenues, including alternative distance measures, efficient algorithms in distributed processing frameworks, and investigations into conditional lower bounds in computational problems, all aimed at enhancing understanding and application of these principles in both theoretical and practical domains [11, 1, 12, 5].The following sections are organized as shown in Figure 1.

2 Background and Preliminary Concepts

2.1 Theoretical Frameworks in Computational Complexity

The study of computational complexity is underpinned by theoretical frameworks that define the limits of efficient computation and elucidate the challenges of complex problems. Fine-grained complexity is pivotal in this regard, providing precise bounds and enhancing the understanding of distributed systems, where graph parameters inform the creation of universally optimal algorithms [4]. The efficient handling of large datasets poses significant challenges, with innovations like the minimum degree algorithm offering heuristic solutions for sparse linear systems [8]. Massively parallel algorithms further address these challenges by optimizing communication and memory use in large-scale graph problems [13].

Graph representations are crucial in fields like molecular engineering, where accurate factorization of weighted graphs is necessary [14]. The exploration of distance closures in complex networks highlights the role of theoretical frameworks in network analysis, revealing isomorphisms between fuzzy and distance graphs [5]. In dynamic graphs, traditional methods often require polynomial update times, limiting their use [15]. Recent advances in data structures and sensitivity oracles have facilitated efficient shortest path calculations amid multiple edge failures [16].

Quantum complexity theory introduces additional challenges, particularly in high-dimensional spaces where classical algorithms fall short. Although promising, quantum algorithms still struggle to outperform brute-force search in these dimensions [17]. The rank-finding conjecture offers insights into the fine-grained hardness of approximation problems [18]. Recent research also emphasizes reducing space complexity for tackling larger instances of problems like the Subset Sum, previously considered intractable [19]. The dynamic k-mismatch problem, involving efficient computation of Hamming distances, underscores the need for effective computational methods [20].

The Balanced Separator problem, which seeks a balanced cut in low-conductance graphs, highlights frameworks addressing complex partitioning challenges [21]. The complexity of identifying multiple kings in tournaments, requiring (n^2) queries even for randomized algorithms, exemplifies inherent difficulties in certain graph problems [22]. These frameworks collectively advance computational complexity research, introducing methodologies that enhance our understanding of complex problems, from graph algorithms to quantum computing. This includes developing new algorithmic tools for well-connected graphs, improved methods for discretized Ricci curvatures, and significant contributions to the dynamic longest increasing subsequence problem and the 3SUM conjecture, illustrating the intricate interconnections within theoretical computer science and how advancements in one domain can catalyze breakthroughs in others [11, 23, 24, 7, 8].

2.2 Energy Efficiency in Computational Models

Energy efficiency is increasingly critical in computational models, especially in distributed systems and large-scale computations. Theoretical computer science emphasizes optimizing energy usage, as evidenced by low-energy deterministic distributed algorithms, which reduce energy consumption by allowing nodes to remain inactive during non-essential computations, thereby enhancing scalability and efficiency [6]. The challenges of super-linear complexities in simple graph problems, which require significant communication in the CONGEST model, underscore the need for energy-efficient solutions [25]. Innovative frameworks that optimize communication and computation are essential to reducing energy consumption.

In dynamic database environments, efficient query evaluation amidst updates and integrity constraints has led to new approaches to computational efficiency [26]. Traditional methods, which often re-run static algorithms or naively recalculate distances after updates, are inefficient, particularly in dynamic scenarios like the dynamic k-mismatch problem [20]. The integration of frameworks such as the multiplicative-weights framework with shortest-path computations simplifies processes like finding balanced sparse cuts, contributing to energy-efficient computational models [21]. These frameworks not only enhance computational efficiency but also play a pivotal role in advancing theoretical computer science by addressing the energy constraints inherent in large-scale and distributed computations.

In examining the intricacies of computational complexity, it is crucial to understand the implications of the Strong Exponential Time Hypothesis (SETH). Figure 2 illustrates the hierarchical structure of SETH and its ramifications across various computational problems. This figure categorizes the impact of SETH on Approximate Nearest Neighbor (ANN) problems, dynamic Longest Increasing Subsequence (LIS) problems, the partition problem, and Fréchet distance approximation. Furthermore, it delineates SETH's role in complexity theory and quantum computing. Each category is meticulously divided into subcategories, emphasizing specific challenges and implications for algorithm development and the broader landscape of computational complexity. This visual representation not only aids in comprehending the multifaceted nature of SETH but also enhances the narrative flow of this review, providing a structured overview of its significance in contemporary computational research.

3 Strong Exponential Time Hypothesis (SETH)

3.1 Implications of SETH for Approximate Nearest Neighbor (ANN) Problems

The Strong Exponential Time Hypothesis (SETH) profoundly influences the complexity of Approximate Nearest Neighbor (ANN) problems by highlighting their inherent hardness. The linkage between ANN and the Shortest Vector Problem (SVP) suggests that if SETH holds, the complexities of solving ANN mirror those of SVP, reflecting similar structural challenges [27]. This connection underscores SETH's broader implications for computational geometry, where ANN's complexity aligns with classical conjectures like the kSUM conjecture [28].

A notable illustration of SETH's impact is the c-Approximate Near Neighbors problem (c-ANN), which involves preprocessing curves to determine if a query curve has a near neighbor within a specific Fréchet distance. Constructing efficient data structures for this task is formidable, with SETH suggesting that sub-exponential time complexity is likely unattainable [29].

Quantum algorithms, while promising, reveal limitations consistent with SETH, indicating that ANN-related problems may not benefit significantly from quantum techniques, reinforcing the hypothesis's constraints on computational efficiency [30]. This aligns with the notion that repetitive data structures can be exploited for reduced running times, yet these improvements remain bounded by SETH's theoretical limits [31].

3.2 SETH and Dynamic Longest Increasing Subsequence (LIS) Problems

The interaction between SETH and dynamic Longest Increasing Subsequence (LIS) problems is critical in computational complexity, particularly concerning the theoretical limits of maintaining LIS amid dynamic changes. The dynamic LIS problem, involving updates to the LIS of a sequence as elements change, presents substantial computational hurdles. Research by Gawrychowski et al. [7]

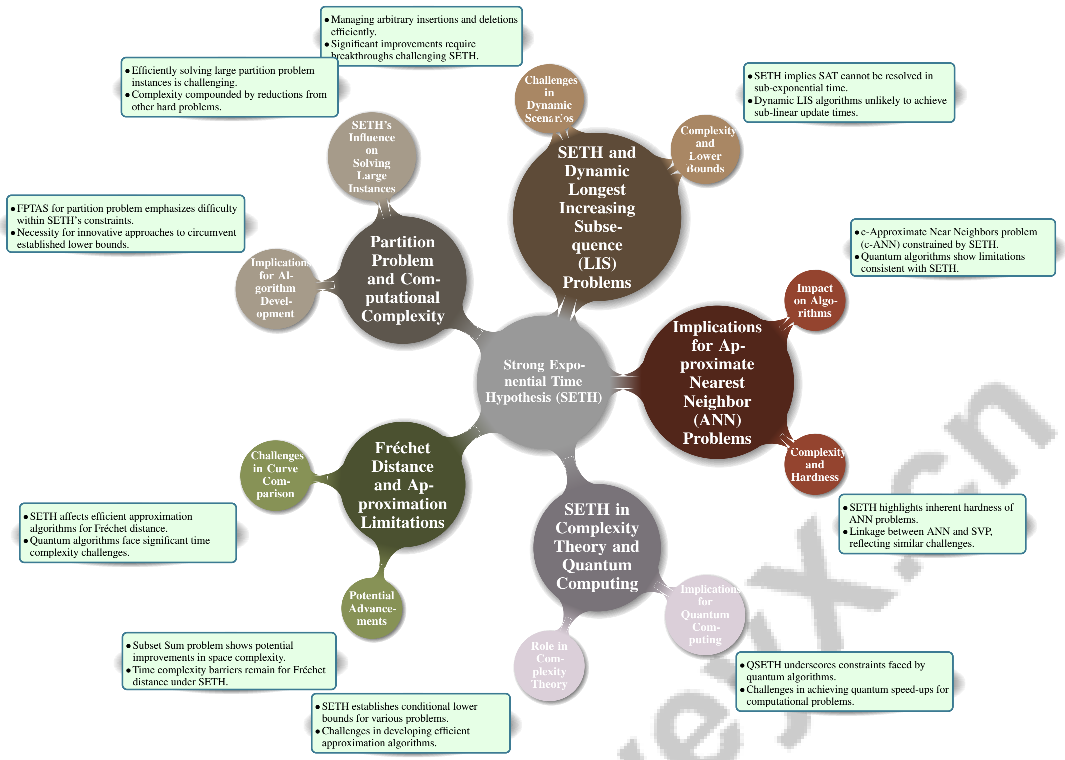

Figure 2: This figure illustrates the hierarchical structure of the Strong Exponential Time Hypothesis (SETH) and its implications across various computational problems. It categorizes the impact of SETH on Approximate Nearest Neighbor (ANN) problems, dynamic Longest Increasing Subsequence (LIS) problems, the partition problem, Fréchet distance approximation, and its role in complexity theory and quantum computing. Each category is further divided into subcategories, highlighting specific challenges and implications on algorithm development and computational complexity.

highlights conditional lower bounds that emphasize the difficulty of achieving efficient updates in dynamic LIS scenarios.

SETH implies that problems like SAT cannot be resolved in sub-exponential time, extending this influence to dynamic LIS complexity. Conditional lower bounds indicate that algorithms for dynamic LIS maintenance are unlikely to achieve sub-linear update times, specifically less than $O(n^{1/2-\epsilon})$ or $O(n^{1/3-\epsilon})$ for weighted elements or 1D queries, respectively [7, 1, 30, 9].

The complexity of dynamic LIS is further heightened by the need to manage arbitrary insertions and deletions efficiently. Gawrychowski et al. [7] assert that significant improvements in update times would require breakthroughs that challenge the current understanding of SETH, indicating the problem's difficulty is deeply rooted in the theoretical constraints imposed by this hypothesis.

3.3 Partition Problem and Computational Complexity under SETH

SETH profoundly affects the computational complexity of the partition problem, which involves dividing a set into two subsets with equal sums. Under SETH, efficiently solving large instances is inherently challenging, as existing methods often fall short due to the hypothesis's constraints [32]. This complexity is compounded by reductions from other hard problems, like k-Orthogonal Vectors, to related issues such as the colorful Ka,b detection problem, establishing SETH-hardness and underscoring the problem's computational intensity [33].

Examining the partition problem under SETH reveals broader implications for computational efficiency. Conditional lower bounds from the All-Pairs Shortest Paths (APSP) conjecture suggest that no algorithm can support queries and updates in time $O(n^{1/2})$ and $O(n^{1/3})$ for dynamic LIS problems, further illustrating SETH's constraints on efficient solutions [7].

Recent research highlights significant theoretical obstacles in devising efficient algorithms for the partition problem, as SETH imposes fundamental limitations on achieving groundbreaking solutions. The best-known Fully Polynomial-Time Approximation Scheme (FPTAS) for the partition problem operates in $O(n + 1/\epsilon)$ time, optimal up to a polylogarithmic factor, emphasizing the problem's inherent difficulty within SETH's constraints [1, 32, 20, 12]. The interplay between SETH and the partition problem highlights the necessity for innovative approaches to circumvent established lower bounds, pushing the boundaries of computational feasibility within the current theoretical framework.

3.4 Fréchet Distance and Approximation Limitations under SETH

The Fréchet distance, a crucial measure for comparing curves, faces notable approximation challenges under SETH. Determining the Fréchet distance between two curves is particularly affected by SETH's limitations, suggesting that efficient approximation algorithms may be inherently difficult to develop [34]. This difficulty is compounded by the structural complexity of continuous curves, complicating the development of algorithms capable of handling approximate queries effectively [29].

SETH's implications extend beyond classical computation, as demonstrated by the Quantum Strong Exponential-Time Hypothesis (QSETH), which posits that even quantum algorithms face significant time complexity challenges for problems like Edit Distance, necessitating $W(n^{1.5})$ time [30]. This trend indicates that both classical and quantum computational paradigms confront barriers in efficiently solving curve similarity problems like the Fréchet distance.

Advancements in other areas, such as the Subset Sum problem, illustrate potential improvements in space complexity, offering insights into tackling large problem instances more effectively [19]. However, these advancements do not necessarily overcome the time complexity barriers imposed by SETH on the Fréchet distance problem.

3.5 SETH in Complexity Theory and Quantum Computing

SETH is a pivotal conjecture in complexity theory, asserting the impossibility of significantly accelerating the satisfiability analysis of CNF formulas beyond exhaustive search methods. This hypothesis has catalyzed extensive research in fine-grained complexity, leading to tight lower bounds for numerous problems in the class P, where establishing unconditional lower bounds remains challenging. In quantum computing, QSETH facilitates the translation of quantum query lower bounds on black-box problems into conditional quantum time lower bounds for various BQP problems. For instance, employing the QSETH framework, researchers have derived a conditional quantum time lower bound of $(n^{1.5})$ for the Edit Distance problem, enhancing our understanding of the limitations and potential of quantum algorithms relative to classical complexity assumptions [30, 35, 17]. SETH posits that certain problems, particularly those related to SAT, cannot be solved in sub-exponential time, establishing a fundamental limit on computational efficiency with profound implications for both classical and quantum algorithm development.

In complexity theory, SETH is instrumental in establishing conditional lower bounds for various computational problems. The complexity of approximating the Fréchet distance intricately links to SETH, suggesting that efficient approximation algorithms may be inherently challenging to develop [34]. This limitation extends to geometric problems, such as the Approximate Nearest Neighbor (ANN) problem, where achieving efficient algorithms under the norm remains unlikely without significant breakthroughs in understanding the underlying hardness assumptions [27].

The implications of SETH transcend classical computation. In quantum computing, QSETH underscores the constraints faced by quantum algorithms in solving problems like Edit Distance, which require substantial time complexity [30]. This highlights the broader challenge of achieving quantum speed-ups for computational problems, as explored in the context of the 3SUM problem [35]. Despite the potential for quantum computing to provide significant advantages, SETH and its quantum counterpart suggest that certain problems may remain resistant to efficient quantum solutions.

The exploration of dynamic algorithms further illustrates the impact of SETH on complexity theory. The conditional lower bounds associated with dynamic problems, such as those related to the All-Pairs Shortest Paths (APSP) and Orthogonal Vectors (OMv) conjectures, emphasize the inherent difficulty of achieving efficient updates and queries in dynamic scenarios [7]. The introduction of APSP solvers in distributed systems, such as Apache Spark, demonstrates the potential for leveraging distributed processing to address these challenges, albeit within SETH's constraints [36].

4 3SUM Conjecture and Its Implications

The 3SUM conjecture provides a pivotal framework in computational theory, defining lower bounds for problems in dynamic data structures, graph algorithms, and string matching. It posits the absence of truly subquadratic algorithms for these problems, thus highlighting their computational hardness and guiding optimal algorithm design [24, 37, 9, 1, 10]. A thorough understanding of the 3SUM problem involves exploring its formal definition and various emerging variants, which illuminate its core aspects and extensive applicability across domains.

4.1 3SUM Problem Definition and Variants

The 3SUM problem involves finding three numbers a, b, and c from a set of n real numbers that sum to zero, i.e., $a + b + c = 0$ [24]. This problem is central to computational complexity and establishing conditional lower bounds in multiple domains [1]. The classical 3SUM problem is conjectured to require $\Theta(n^2)$ time, a pivotal factor in deriving lower bounds for dynamic data structures and graph problems [9]. The Quantum-3SUM-Conjecture extends this to quantum computing, suggesting the nonexistence of sublinear quantum algorithms for the problem [35].

Variants of 3SUM address distinct computational challenges. For instance, detecting Abelian and additive square factors in strings involves identifying substrings that are permutations or sum to the same value [10]. The dynamic 3SUM problem, relevant in dynamic graph problems, maintains solutions amid updates, crucial for preserving approximate shortest paths during edge modifications [38]. The conjecture's influence extends to computational geometry, affecting algorithm development for tasks like approximate nearest neighbors under specific norms [27], and highlights challenges in distributed computing, such as the lack of efficient MapReduce algorithms for 3SUM [12].

4.2 Algorithmic Approaches and Innovations

The 3SUM problem has catalyzed algorithmic innovations aimed at enhancing efficiency and tackling related computational challenges. Notably, a decision tree with depth $O(n^{3/2} \log n)$ and subquadratic algorithms surpass traditional $O(n^2)$ complexity, facilitating more efficient solutions [24]. These methods typically involve sorting and partitioning the input set for efficient sum searching [24].

3SUM-hardness has been demonstrated in related problems, such as detecting odd half-length Abelian squares and computing centers of Abelian square factors, extending 3SUM reductions' applicability [10]. New reductions from 3SUM to offline SetDisjointness and SetIntersection exemplify innovative strategies leveraging 3SUM complexity to address other challenging problems [9].

In distributed computing, new MapReduce algorithms optimizing memory and machine usage significantly advance solving fundamental problems like 3SUM in large-scale environments [12]. These algorithms enhance scalability and performance in distributed systems. Quantum algorithm exploration has yielded promising results, aiming to generalize techniques for broader geometric problems and exploit quantum speedups for 3SUM and its variants [37]. Conditional hardness results for problems like finding a $(1 + \epsilon)$-approximate nearest neighbor under ℓ_∞ norm illustrate 3SUM's broader implications in computational geometry, where efficient algorithms depend on hypotheses like SETH [27].

Advancements in algorithmic techniques around 3SUM challenge the conjectured optimal time complexity, enhancing understanding across various computational fields. These innovations have led to subquadratic algorithms for 3SUM, improved lower bounds for dynamic data structures, and new reductions in graph theory and string processing, driving progress in theoretical and applied computer science [24, 9, 1, 12, 10].

4.3 Significance in Computational Geometry

The 3SUM problem's significance in computational geometry lies in its foundational role in establishing conditional lower bounds for various geometric problems. The 3SUM conjecture is a critical tool in computational complexity, providing a framework to assess problem hardness through well-established reductions [1]. It is particularly influential in computational geometry, aiding in understanding geometric configurations' complexity and solving related problems efficiently.

The identification of 3SUM-hardness in detecting Abelian and additive square factors illustrates the intricate connections between 3SUM and geometric problem-solving, establishing conditional lower bounds and guiding efficient algorithm development [10]. Exploring quantum speedups for geometric 3SUM-hard problems emphasizes 3SUM's relevance in computational geometry, as quantum computing offers potential pathways to overcome classical limitations [37]. However, as noted by Buhrman et al. [35], the classical 3SUM problem and its variants remain central to understanding quantum speedup limits.

The necessity for parallel approaches to problems like All-Pairs Shortest Paths (APSP) in computational geometry is underscored by the increasing size of input graphs in practical applications [36]. Developing algorithms that efficiently manage large-scale data is crucial, and insights from 3SUM studies are invaluable in this context. Furthermore, advancements in algorithmic approaches to the 3SUM problem demonstrate significant improvements over previous methods, challenging the conjectured optimal time complexity and opening avenues for future research into refining these algorithms and exploring their applications in other geometric and algorithmic complexity problems [24].

5 All-Pairs Shortest Paths (APSP)

5.1 Dynamic and Decremental APSP

Dynamic and decremental APSP are central in graph algorithms, focusing on maintaining shortest paths amid frequent graph changes like edge insertions and deletions. Recent advances have introduced algorithms enhancing both theoretical insights and practical applications. For instance, a new algorithm for dynamic spanners exhibits sub-linear recourse during edge updates, while another integrates shortest-path computations with multi-commodity flows for efficient sparse cut identification. A dynamic APSP data structure processes edge modifications in nearly linear time, enabling rapid approximate distance queries. Notably, improvements in maintaining APSP under adaptive adversaries have led to better update times for exact and approximate distances in directed graphs, refining methodologies and elevating performance in dynamic graph contexts [21, 38, 39, 40].

Significant progress in decremental APSP algorithms has introduced new trade-offs enhancing approximation guarantees for both weighted and unweighted graphs, advancing state-of-the-art techniques [41]. These innovations are vital for applications requiring efficient management of decremental changes in large-scale networks, where accurate shortest path distances must be preserved [40].

Distributed computing has further propelled dynamic APSP advancements. Huang's method [42] achieves exact weighted APSP in $\tilde{O}(n^{5/4})$ rounds, significantly outperforming naive $O(m)$-time algorithms, highlighting distributed algorithms' scalability in APSP computations.

Deterministic techniques for constructing successive blocker sets, as explored by Agarwal et al. [43], enable shortest cycles computation in $\tilde{O}(n)$ rounds, showcasing deterministic strategies' efficacy in optimizing dynamic graph algorithms.

Additionally, fast matrix multiplication techniques combined with bridging sets enhance efficiency in solving APSP problems [44]. These methods leverage matrices' algebraic properties to expedite shortest path computations, reducing computational overhead in dynamic scenarios.

In quantum computing, integrating classical matrix multiplication techniques with quantum search algorithms has demonstrated faster APSP computation times [45]. This approach benefits from quantum algorithms' superior search and computation capabilities, particularly in structured graphs [45].

Overall, advancements in dynamic and decremental APSP algorithms signify substantial progress in addressing computational challenges within dynamic graph environments. These include efficient algorithms maintaining approximate spanners and facilitating quick access to shortest paths during edge modifications. For instance, one algorithm achieves an update time of $O(m^{1+o(1)})$ while

AI-generated, for reference only.

maintaining an $n^{o(1)}$-approximate spanner, improving recourse efficiency when insertions exceed deletions. Another method allows dynamic approximate distance calculations with a total update time of $|E|^{1+o(1)}$ and constant-time query access. Additionally, new trade-offs between approximation and running time have been established, enabling quicker updates for both weighted and unweighted graphs, broadening APSP algorithms' applicability across various fields, including network design, routing, and real-time traffic analysis [21, 39, 40, 38, 41]. The interplay of deterministic and distributed strategies, alongside quantum techniques integration, underscores ongoing innovations in this domain, propelling further research and development in graph algorithms.

5.2 Distributed and Parallel Approaches

Distributed and parallel approaches to APSP are crucial for efficiently managing large-scale graphs and dynamic network environments. These methodologies exploit graph computations' inherent parallelism to significantly reduce computational and communication overheads, enhancing scalability and performance in distributed processing environments. They leverage advanced algorithms tailored for massively parallel computation (MPC) models and MapReduce frameworks, achieving sublogarithmic-time solutions for fundamental graph problems, including APSP and spanner construction. By optimizing memory usage across multiple machines and effectively partitioning tasks, these strategies improve efficiency while ensuring effectiveness across various network topologies, enabling faster processing in real-world applications [4, 12, 13].

A notable advancement in this area is Huang's method [42], which integrates multiple distributed algorithms to compute shortest paths efficiently while addressing edge weights and communication constraints. This exemplifies how distributed computing can facilitate scalable APSP solutions by optimizing resource allocation and minimizing communication delays.

Dang's theoretical analysis [46] evaluates distributed algorithms' efficiency by assessing the fitness evaluations necessary for accurate shortest path computations for all pairs. This analysis underscores distributed strategies' effectiveness in reducing computational complexity and accelerating convergence in APSP calculations.

Evald introduces three innovative data structures for decremental APSP in directed graphs—one for exact distances, one for (1 +)-approximate distances, and a randomized structure effective against adaptive adversaries [40]. These structures enhance distributed APSP algorithms' robustness and efficiency, particularly in dynamic scenarios with frequent edge modifications.

The integration of parallel computing techniques, as demonstrated by Zwick [44], involves calculating distance products of matrices representing the graph. This iterative refinement of shortest path estimates significantly boosts computational throughput, offering a promising avenue for optimizing parallel APSP computations.

Future research, as suggested by Agarwal et al., could explore the feasibility of achieving a deterministic algorithm aligning with the $\tilde{O}(n)$ randomized bound for APSP while optimizing current methodologies. This highlights ongoing efforts to develop more efficient and scalable solutions for distributed and parallel APSP algorithms, particularly within modern frameworks such as MapReduce and Apache Spark, which face challenges in balancing parallelism with system constraints. Recent advancements include algorithms optimizing memory and machine utilization and dynamic approaches adapting to edge updates, emphasizing these innovations' importance for large-scale graph processing in practical applications [21, 36, 39, 12, 38].

5.3 Approximation and Quantum Algorithms

Examining approximation and quantum algorithms for APSP is crucial for enhancing computational efficiency, particularly in large-scale and dynamic graph contexts. Recent advancements feature algorithms providing significant trade-offs between approximation accuracy and update times for both weighted and unweighted graphs, achieving faster updates than previous state-of-the-art methods. For instance, randomized algorithms now offer (2 +)-approximation with total update times as low as $\tilde{O}(m^{1/2}n^{3/2})$, while improvements in fully dynamic settings have led to worst-case update times of $\tilde{O}(n^{2+5/7})$ for maintaining distance matrices amid vertex insertions and deletions. These innovations tackle dynamic graph structures' complexities and contribute to computational graph analysis's theoretical foundations [47, 48, 49, 41, 50]. Such advancements are vital for improving

APSP computations' scalability and performance, where traditional methodologies often encounter inherent limitations.

Noteworthy improvements in approximation algorithms for APSP have emerged. Dory et al. [41] present decremental APSP algorithms that exceed existing methods in both update times and approximation ratios, offering effective solutions for maintaining shortest paths in graphs facing edge deletions. This underscores innovative approaches' potential to significantly enhance APSP computations.

The integration of distributed computing techniques has also been instrumental in advancing APSP solutions. Huang [42] proposes a distributed algorithm that efficiently computes exact weighted shortest paths, optimizing the communication rounds required while ensuring high accuracy in results. This exemplifies leveraging both hop length and weighted distance in determining shortest paths, facilitating efficient communication in distributed environments [51].

In quantum algorithms, Nayebi et al. [45] present a quantum algorithm achieving a time complexity of $O(n^{2.5} - \epsilon)$ for APSP in geometrically weighted graphs, illustrating quantum computing's potential to expedite APSP computations. However, quantum algorithms' application to APSP may be limited by fundamental complexity constraints, as demonstrated by Wang et al., who indicate no quantum speedup for computing eccentricities and APSP [52].

Moreover, developing efficient algorithms for the Min-Plus product, as described by Williams [53], yields the first truly subcubic algorithm for less structured matrices, significantly enhancing computational efficiency for related challenges, including APSP. This approach exemplifies specialized algorithms' potential to improve APSP solutions by exploiting algebraic properties.

Furthermore, introducing complementary versions of graph centrality problems, such as CoDiameter, CoMedian, and CoRadius, illustrates subcubic equivalences with APSP, offering new insights into APSP-related challenges' complexity [54]. These innovations emphasize APSP's interconnected nature with broader computational issues, driving ongoing research and development in graph algorithms.

6 Orthogonal Vectors Problem

6.1 Overview of the Orthogonal Vectors Problem

The Orthogonal Vectors (OV) problem is a core challenge in theoretical computer science, involving the determination of orthogonality between any two vectors within a set of n binary vectors in $\{0, 1\}^{O(\log n)}$ [55]. This problem is integral to fine-grained complexity theory, underpinning our understanding of computational limits for related problems like the 3SUM problem and All-Pairs Shortest Paths (APSP) [12]. The complexity of the OV problem stems from the need to efficiently identify orthogonal pairs among numerous vectors, with current algorithms achieving a time complexity of $O(n^2 \log n)$. Conditional lower bounds suggest that solving the OV problem in subquadratic time, such as $n^{1.99}$, is improbable [56, 55, 19, 57]. The exponential increase in vector space complexity with dimensional growth further challenges traditional methods.

Recent advancements have introduced algorithms that improve the efficiency of solving the OV problem. For instance, constant round algorithms for OV leverage parallelism in distributed computing to reduce computational overhead in large-scale data processing [12]. The OV problem also significantly impacts network analysis, particularly in approximating network diameter, which is crucial for evaluating maximum distances between nodes [58]. Moreover, the OV problem intersects with classical problems like the Subset Sum problem, highlighting the interconnected nature of computational challenges in theoretical computer science [57].

6.2 Reductions and Algorithmic Approaches

The OV problem is pivotal in fine-grained complexity, with its complexity informing the computational limits of various related problems. Two innovative reduction frameworks to OV have emerged, clarifying its interrelationships with other computational challenges [55]. These frameworks establish conditional lower bounds for numerous problems by leveraging OV's complexity as a baseline.

Reduction techniques often involve transforming problems into high-dimensional vector spaces to identify orthogonal pairs, essential for developing efficient algorithms amidst exponential vector dimension growth. Constant round algorithms exemplify the potential for optimizing computational overhead in large-scale data processing environments [12]. The OV problem is also linked to computing or approximating the diameter in sparse graphs, requiring several rounds [58]. This connection underscores OV's broader implications in network analysis, as understanding diameter is vital for evaluating network properties.

Algorithmic approaches to the OV problem increasingly focus on distributed and parallel computing techniques to enhance efficiency. Innovations, such as advancements in Online Boolean Matrix-Vector Multiplication (OMV), are critical for addressing OV complexities and understanding efficient computation boundaries. These developments facilitate scalable solutions to intricate challenges like OV while improving performance in applications such as independent set detection and matrix multiplication in distributed frameworks like MapReduce. Notably, a new randomized algorithm for OMV significantly reduces computation time, revealing deeper connections between matrix-vector multiplication and foundational computational complexity problems [56, 12].

6.3 Applications and Implications

The OV problem has substantial applications across various domains in theoretical computer science, serving as a benchmark for understanding the computational complexity of high-dimensional data problems. It establishes conditional lower bounds for numerous computational tasks, deriving hardness results for problems like APSP and the Diameter problem in sparse graphs, where substantial computational resources are essential for efficient solutions [58].

In network analysis, the OV problem is crucial for accurately estimating key characteristics such as diameter and radius, representing maximum and minimum distances between nodes. These parameters are vital for assessing network performance and connectivity, influencing data transmission efficiency and network robustness. Various algorithms have been developed to approximate these measures in both static and dynamic graphs, significantly impacting real-time network monitoring and analysis. Recent advancements have led to subquadratic approximation algorithms that enhance computational efficiency, thereby improving the management and evaluation of complex networks [59, 60, 61, 58, 5]. The task of determining orthogonal pairs among vectors parallels assessing maximum distances between nodes in a network, providing insights into network efficiency and reliability.

Moreover, the implications of the OV problem extend into quantum computing, where adapting classical reductions to quantum settings offers new perspectives on computational complexity [30]. This cross-disciplinary approach highlights the potential for quantum algorithms to tackle the computational challenges posed by OV, although significant barriers remain in achieving substantial quantum speedups.

Exploring the OV problem also informs algorithmic development in classical problems such as the Subset Sum problem, providing a framework for understanding the limits of efficient computation in high-dimensional spaces. The intersection of these problems underscores the interconnected nature of computational challenges, where advancements in one area can catalyze breakthroughs in another.

7 Fine-Grained Reductions and Dynamic Lower Bounds

7.1 Conceptual Framework of Fine-Grained Reductions

Fine-grained reductions are fundamental in complexity theory, enabling the transformation of problems while maintaining computational hardness, thereby establishing conditional lower bounds. For example, reductions from the 3SUM problem to Abelian and additive squares highlight the complexity of these problems through combinatorial techniques [10]. These reductions are particularly useful in approximating counting problems, where decision oracles streamline complexity analysis [62].

The framework extends to dynamic algorithms, as demonstrated by maintaining a (1 + ,)-spanner on undirected graphs during edge modifications, showcasing the adaptability of reductions in dynamic data structures [63]. Limitations in dynamic fractional cascading within the pointer machine model further reveal constraints in dynamic computational models [64]. Boolean matrix structures and properties of the Orthogonal Vectors problem illustrate how fine-grained reductions facilitate faster computations [56]. Applications from planar graph constructions establish polynomial lower bounds for dynamic Longest Increasing Subsequence problems, particularly in weighted and 1D-query variants, demonstrating innovative uses of reductions in deriving complexity bounds [7].

Additionally, the interplay between approximate counting and decision problems in hypergraphs underscores the theoretical depth of fine-grained reductions [65]. In dynamic graph contexts, maintaining vertex sparsifiers with subpolynomial update times exemplifies the practical applications of these reductions in optimizing dynamic algorithms [15]. The development of space-efficient algorithms, such as achieving $O^*(2^{0.246n})$ for the Subset Sum problem, highlights how reductions enhance computational efficiency [19]. Furthermore, trade-off algorithms allowing flexible update and query times, as seen in the dynamic k-mismatch problem, illustrate the innovative solutions enabled by fine-grained reductions [20]. Recent advancements include new upper bounds for better approximation factors with linear preprocessing time [29]. Collectively, these developments underscore the critical role of fine-grained reductions in advancing theoretical computer science and exploring the limits of efficient computation.

7.2 Innovative Data Structures for Dynamic Scenarios

Innovative data structures for dynamic scenarios are crucial for addressing challenges in dynamic graph environments characterized by frequent updates. Recent advancements have introduced dynamic approximation algorithms that utilize randomized techniques and careful sampling to achieve efficient updates, ensuring accurate estimates of graph parameters [59]. These algorithms optimize dynamic systems by preserving critical graph properties amid data changes.

The DynamicSpanner algorithm is a notable innovation, enhancing efficiency by maintaining low congestion in embedding paths, thereby preserving spanner properties during updates and optimizing the balance between computational overhead and accuracy [39]. This dynamic maintenance is particularly valuable in large-scale networks, where efficient path maintenance is crucial for performance and reliability.

These advancements highlight the importance of designing algorithms that adapt to changing data environments while minimizing computational costs. By employing innovative randomized techniques and ensuring low congestion through advanced vertex sparsifiers, these data structures effectively manage dynamic graph problems, including approximate all-pairs shortest paths (APSP) and single-source shortest paths (SSSP). This progress enhances theoretical computer science and facilitates practical applications in flow-routing algorithms and large-scale graph computations, exemplified by implementations on distributed systems like Apache Spark that efficiently handle graphs with over 200,000 vertices [11, 36, 15].

7.3 Applications and Examples of Fine-Grained Reductions

Fine-grained reductions are pivotal in establishing precise connections between the complexities of various computational problems, aiding in understanding their inherent difficulty and facilitating efficient algorithm development. A notable application is in the classification of parity problems, where complexity relationships are meticulously detailed, providing insights into the equivalence classes of computational problems [66]. The Orthogonal Vectors (OV) problem serves as a cornerstone in fine-grained complexity, with its equivalence classes expanded to include new tools for establishing lower bounds [55]. For example, reductions from the OV problem to the diameter approximation problem demonstrate that both problems share similar complexity, highlighting their interconnected nature [58].

In graph theory, fine-grained reductions significantly impact problems involving matrix operations and the Min-Plus product. The development of truly subcubic algorithms for the Min-Plus product exemplifies the potential of fine-grained reductions to enhance computational efficiency in graph-related problems [53]. These advancements are crucial for optimizing algorithm performance in large-scale graph environments.

The application of fine-grained reductions extends to optimizing dynamic algorithms, where new techniques assess performance by measuring update times and the quality of maintained structures, such as spanners [63]. These techniques improve the adaptability and efficiency of algorithms in dynamic settings, where data frequently changes. Additionally, the introduction of space-efficient reductions in problems like the Subset Sum problem, leveraging refined algorithms for detecting orthogonal pairs, showcases innovative uses of fine-grained reductions to solve complex problems with limited resources [57]. This approach enhances space efficiency and provides a framework for tackling high-dimensional problems.

7.4 Future Directions and Open Questions

Exploring fine-grained reductions offers numerous opportunities for advancing our understanding of computational complexity, with several promising directions and open questions. One area ripe for exploration involves further optimizations of algorithms related to combinatorial optimization problems, such as partition problems [32]. These optimizations could broaden the applicability of current methods, enhancing their practical utility.

In dynamic graph problems, refining algorithms to improve resilience to edge deletions and exploring additional applications of well-connected graphs remain critical areas for future research [11]. These advancements could yield more robust solutions capable of efficiently managing dynamic changes in network structures. The potential for quantum advantages in weighted networks and other distance parameters presents another intriguing direction for future work. This includes refining quantum protocols to enhance efficiency, particularly concerning eccentricities and all-pairs shortest paths (APSP) problems [52]. Such research could uncover new quantum speedups that challenge existing complexity assumptions.

Further optimizations of the target-(min,max)-product problem and the exploration of additional reductions that enhance APSP algorithm performance are also promising areas for investigation [67]. These efforts could lead to more efficient solutions for various graph-related challenges. Refining distributed approximation algorithms, particularly those tailored to different network topologies, represents another vital area for future research. Extensions to other combinatorial optimization problems could expand these algorithms' applicability, providing scalable solutions for complex network environments [68].

The Quantum Strong Exponential Time Hypothesis (QSETH) framework offers fertile ground for exploring other computational problems, particularly in understanding compression-oblivious properties [30]. Future work in this area could lead to significant breakthroughs in quantum computing, enhancing our understanding of its potential and limitations. Further optimizations of reductions related to the 3SUM problem and their applicability to other computational complexity problems remain open questions [9]. These investigations could reveal new insights into the interconnected nature of computational problems and the potential for cross-problem optimizations.

Lastly, exploring further optimizations of the emulator construction process and its application to other graph problems, including potential extensions to directed graphs, offers another promising research direction [69]. These extensions could lead to more efficient algorithms capable of handling a wider array of graph scenarios. The exploration of future directions and unresolved questions in fine-grained reductions highlights the evolving landscape of research in this area, revealing a wealth of opportunities to deepen our understanding of computational complexity and enhance its practical applications. Recent advancements, such as parameterized fine-grained reductions, have established a framework for analyzing the intricacies of improvable problems, while innovative algorithms for distributed processing systems like MapReduce demonstrate the potential for efficient solutions to fundamental problems, including Matrix Multiplication and All Pairs Shortest Paths. Furthermore, optimal hardness results for approximating linear systems and dynamic algorithms for maintaining shortest path information in graphs underscore the significance of fine-grained techniques in addressing both theoretical challenges and real-world computational tasks [18, 70, 12, 63].

8 Subquadratic Algorithms and NP-Hardness

8.1 Deterministic Approaches and Dynamic Graphs

Deterministic subquadratic algorithms are increasingly vital in dynamic graph contexts, where computational efficiency must be preserved despite structural changes like edge insertions and deletions. Challenges persist in maintaining subquadratic performance for problems such as minimum degree ordering and dynamic all-pairs shortest paths (APSP), where speed and accuracy must be balanced amid frequent updates [11, 71, 25, 8, 63]. Recent innovations have achieved constant factor approximations for graph girth in subquadratic time, enhancing network design applications by informing resilience and optimization decisions [72].

These deterministic strategies employ advanced data structures for rapid updates and queries, allowing algorithms to adapt dynamically. Notably, new APSP algorithms use sophisticated techniques like blocker sets and pipelined distance propagation, achieving $\tilde{O}(n^{4/3})$ rounds in the Congest model for arbitrary edge weights, an improvement over previous $\tilde{O}(n^{3/2})$ bounds [73, 74]. This enhances distance computations in large networks, offering broader implications for distributed algorithms.

The intersection of deterministic approaches and fine-grained complexity delineates computational boundaries, as seen in efficient algorithms for maintaining shortest paths and analyzing parameters like diameter and radius. Recent research, grounded in hypotheses like the OMv conjecture and Combinatorial k-Clique hypothesis, suggests significant algorithmic changes are necessary for substantial improvements [11, 70, 63, 59]. Establishing subquadratic bounds helps in understanding problem difficulty and developing algorithms that push computational feasibility limits.

8.2 3SUM-Hardness and Related Problems

3SUM-hardness is pivotal in assessing the computational limits of various problems. The 3SUM conjecture, positing the nonexistence of strongly subquadratic algorithms for the 3SUM problem, sets a benchmark for computational difficulty across related problems [9]. Establishing 3SUM-hardness involves proving that problems cannot be solved in strongly subquadratic time unless the conjecture is refuted, as seen in detecting Abelian and additive square factors [10]. Such reductions are crucial for conditional lower bounds and evaluating algorithm efficiency.

In geometric contexts, the 3SUM conjecture's constraints on classical algorithms underscore barriers to subquadratic solutions [37]. Parameterized complexity further examines fixed-parameter improvements to enhance efficiency, with fine-grained reductions optimizing performance within hardness constraints [70].

The Approximate Nearest Neighbor (ANN) problem exemplifies 3SUM-hardness challenges, particularly in achieving sublinear query times with polynomial preprocessing. Hardness results for ANN suggest breakthroughs are needed to surmount 3SUM-imposed barriers [27]. The significance of 3SUM-hardness extends to problems like k-Orthogonal Vectors and k-SUM, emphasizing the interconnected nature of computational challenges [65].

8.3 Graph Parameters and Subquadratic Time

Research into graph parameters like radius and diameter in relation to subquadratic algorithms is crucial for understanding computational efficiency. While approximation algorithms exist for these parameters, theoretical barriers suggest exact solutions may not be achievable in subquadratic time under accepted assumptions. This research highlights algorithmic limitations and propels the search for innovative methods to redefine algorithmic efficiency in graph theory [72, 71, 8, 60, 54].

A significant contribution is the refined understanding of diameter approximation's relationship with the Orthogonal Vectors problem, which could improve algorithm efficiency in distributed environments [58]. The study of the Fréchet distance and its variants further illustrates complexities in developing subquadratic algorithms for graph problems [34].

Integrating fine-grained complexity techniques into graph parameter studies opens new avenues for subquadratic solutions. By establishing conditional lower bounds and exploring reductions between graph problems and challenges like the 3SUM conjecture, researchers can identify optimization opportunities. Insights from dynamic data structure analyses reveal inherent time complexity limitations, guiding the development of more efficient methods within established frameworks [25, 7, 8, 9, 1].

8.4 Dynamic and Fine-Grained Complexity

Exploring dynamic and fine-grained complexity is vital for developing subquadratic algorithms, revealing the interdependencies between problem structures and computational efficiencies. Recent advancements in fine-grained complexity, through parameterized reductions, connect various problem complexities, facilitating running time improvements and hardness results. A truly subquadratic fixed-parameter algorithm for the orthogonal vectors problem exemplifies how understanding these complexities enhances algorithmic performance across applications, from linear equation solving to distributed processing [18, 8, 70, 12].

Dynamic complexity focuses on maintaining solutions amid data changes, crucial in real-world applications. Fine-grained complexity establishes precise computational limits, understanding problem nuances based on difficulty. Recent advancements suggest potential for improved approximation guarantees and methods for dynamic graphs, where maintaining efficient solutions amid changes is critical [66].

The pursuit of subquadratic algorithms is informed by techniques enhancing online computations, like matrix-vector multiplication, foundational for further dynamic problem research [56]. Parameterized fine-grained reductions provide a framework for understanding fixed-parameter improvements and traditional complexity classes, offering new optimization avenues [70].

Limitations persist, such as generalizing methods to graphs with negative weights or specific topologies where blocker sets may not cover all paths [43]. Future research could refine these algorithms, explore deterministic approaches, and extend methods to other NP-complete problems [57].

9 Conclusion

This survey articulates the complex interdependencies among pivotal concepts in theoretical computer science, such as SETH, 3SUM, and APSP, highlighting their collective impact on the evolution of computational complexity. These interconnections are instrumental in establishing conditional lower bounds and elucidating the intrinsic challenges of computational problems. The exploration of parity variants underscores the complexity inherent in problems like the Negative Weight Triangle, revealing that these variations can be equally or more demanding than their original forms. Fine-grained reductions further underscore their significance in complexity theory by transforming approximate counting problems into decision problems with minimal computational overhead.

The survey underscores the critical role of energy efficiency in computational models, especially in distributed systems where achieving near-optimal performance with minimal energy consumption is increasingly vital. The advancements in universally optimal algorithms within hybrid models exemplify significant progress, demonstrating the potential for optimal broadcasting in intricate network environments.

The adaptability of algorithms in dynamic and fine-grained complexity settings is also highlighted. The dynamic k-mismatch algorithm exemplifies significant advancements, providing optimal trade-offs for updates and queries in practical applications. The efficient maintenance of graph parameters amidst dynamic changes is crucial for optimizing real-world performance, with the DynamicSpanner algorithm offering notable improvements for edge-dynamic graphs. Future research is encouraged to focus on optimizing algorithms for weighted graphs and exploring the effects of negative weights in mixed graphs.

The survey concludes by affirming that the proposed data structures achieve tight bounds for approximate near neighbor searching under the continuous Fréchet distance, surpassing previous methodologies and suggesting new research directions. The findings also indicate that APSP and Co-Diameter are subcubic equivalent, with similar equivalences applicable to other centrality problems, thereby advancing the comprehension of their computational complexities.

References

[1] Isaac Goldstein, Tsvi Kopelowitz, Moshe Lewenstein, and Ely Porat. How hard is it to find (honest) witnesses?, 2017.

[2] Amir Abboud, Arturs Backurs, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Or Zamir. Subtree isomorphism revisited, 2015.

[3] Yi-Jun Chang, Oren Hecht, and Dean Leitersdorf. Universally optimal deterministic broadcasting in the hybrid distributed model, 2023.

[4] Bernhard Haeupler, David Wajc, and Goran Zuzic. Universally-optimal distributed algorithms for known topologies, 2023.

[5] Tiago Simas and Luis M Rocha. Distance closures on complex networks, 2014.

[6] Udit Agarwal and Vijaya Ramachandran. New and simplified distributed algorithms for weighted all pairs shortest paths, 2018.

[7] Paweł Gawrychowski and Wojciech Janczewski. Conditional lower bounds for variants of dynamic lis, 2021.

[8] Robert Cummings, Matthew Fahrbach, and Animesh Fatehpuria. A fast minimum degree algorithm and matching lower bound, 2020.

[9] Tsvi Kopelowitz, Seth Pettie, and Ely Porat. Higher lower bounds from the 3sum conjecture, 2019.

[10] Jakub Radoszewski, Wojciech Rytter, Juliusz Straszyński, Tomasz Waleń, and Wiktor Zuba. Hardness of detecting abelian and additive square factors in strings, 2021.

[11] A distanced matching game decremental apsp in expanders and faster deterministic algorithms for graph cut problems.

[12] MohammadTaghi Hajiaghayi, Silvio Lattanzi, Saeed Seddighin, and Cliff Stein. Mapreduce meets fine-grained complexity: Mapreduce algorithms for apsp, matrix multiplication, 3-sum, and beyond, 2019.

[13] Amartya Shankha Biswas, Michal Dory, Mohsen Ghaffari, Slobodan Mitrović, and Yasamin Nazari. Massively parallel algorithms for distance approximation and spanners, 2021.

[14] Kristin Sheridan, Joseph Berleant, Mark Bathe, Anne Condon, and Virginia Vassilevska Williams. Factorization and pseudofactorization of weighted graphs, 2021.

[15] Rasmus Kyng, Simon Meierhans, and Maximilian Probst Gutenberg. A dynamic shortest paths toolbox: Low-congestion vertex sparsifiers and their applications, 2023.

[16] Virginia Vassilevska Williams, Eyob Woldeghebriel, and Yinzhan Xu. Algorithms and lower bounds for replacement paths under multiple edge failures, 2022.

[17] Scott Aaronson, Nai-Hui Chia, Han-Hsuan Lin, Chunhao Wang, and Ruizhe Zhang. On the quantum complexity of closest pair and related problems, 2020.

[18] Mitali Bafna and Nikhil Vyas. Optimal fine-grained hardness of approximation of linear equations, 2021.

[19] Tatiana Belova, Nikolai Chukhin, Alexander S. Kulikov, and Ivan Mihajlin. Improved space bounds for subset sum, 2024.

[20] Raphaël Clifford, Paweł Gawrychowski, Tomasz Kociumaka, Daniel P. Martin, and Przemysław Uznański. The dynamic k-mismatch problem, 2022.

[21] Li Chen, Rasmus Kyng, Maximilian Probst Gutenberg, and Sushant Sachdeva. A simple framework for finding balanced sparse cuts via apsp, 2022.

[22] Amir Abboud, Tomer Grossman, Moni Naor, and Tomer Solomon. From donkeys to kings in tournaments, 2024.

[23] Bhaskar DasGupta, Elena Grigorescu, and Tamalika Mukherjee. On computing discretized ricci curvatures of graphs: local algorithms and (localized) fine-grained reductions, 2023.

[24] Allan Grønlund and Seth Pettie. Threesomes, degenerates, and love triangles, 2014.

[25] Keren Censor-Hillel, Seri Khoury, and Ami Paz. Quadratic and near-quadratic lower bounds for the congest model, 2017.

[26] Christoph Berkholz, Jens Keppeler, and Nicole Schweikardt. Answering ucqs under updates and in the presence of integrity constraints, 2017.

[27] Young Kun Ko and Min Jae Song. Hardness of approximate nearest neighbor search under l-infinity, 2020.

[28] Marvin Künnemann and André Nusser. Polygon placement revisited: (degree of freedom + 1)-sum hardness and an improvement via offline dynamic rectangle union, 2021.

[29] Karl Bringmann, Anne Driemel, André Nusser, and Ioannis Psarros. Tight bounds for approximate near neighbor searching for time series under the fréchet distance, 2021.

[30] Harry Buhrman, Subhasree Patro, and Florian Speelman. The quantum strong exponential-time hypothesis, 2019.

[31] Alejandro Cassis, Tomasz Kociumaka, and Philip Wellnitz. Optimal algorithms for bounded weighted edit distance, 2023.

[32] Lin Chen, Jiayi Lian, Yuchen Mao, and Guochuan Zhang. Approximating partition in near-linear time, 2024.

[33] Shuichi Hirahara and Nobutaka Shimizu. Nearly optimal average-case complexity of counting bicliques under seth, 2020.

[34] Kevin Buchin, Tim Ophelders, and Bettina Speckmann. Seth says: Weak fréchet distance is faster, but only if it is continuous and in one dimension, 2018.

[35] Harry Buhrman, Bruno Loff, Subhasree Patro, and Florian Speelman. Limits of quantum speed-ups for computational geometry and other problems: Fine-grained complexity via quantum walks, 2021.

[36] Frank Schoeneman and Jaroslaw Zola. Solving all-pairs shortest-paths problem in large graphs using apache spark, 2019.

[37] J. Mark Keil, Fraser McLeod, and Debajyoti Mondal. Quantum speedup for some geometric 3sum-hard problems and beyond, 2024.

[38] Rasmus Kyng, Simon Meierhans, and Gernot Zöcklein. Bootstrapping dynamic apsp via sparsification, 2024.

[39] Rasmus Kyng, Simon Meierhans, and Gernot Zöcklein. A simple dynamic spanner via apsp, 2024.

[40] Jacob Evald, Viktor Fredslund-Hansen, Maximilian Probst Gutenberg, and Christian Wulff-Nilsen. Decremental apsp in directed graphs versus an adaptive adversary, 2020.

[41] Michal Dory, Sebastian Forster, Yasamin Nazari, and Tijn de Vos. New tradeoffs for decremental approximate all-pairs shortest paths, 2024.

[42] Chien-Chung Huang, Danupon Nanongkai, and Thatchaphol Saranurak. Distributed exact weighted all-pairs shortest paths in $\tilde{o}(n^{5/4})$ rounds, 2017.

[43] Udit Agarwal. Distributed deterministic exact minimum weight cycle and multi source shortest paths in near linear rounds in congest model, 2023.

[44] Uri Zwick. All pairs shortest paths using bridging sets and rectangular matrix multiplication, 2000.

[45] Aran Nayebi and Virginia Vassilevska Williams. Quantum algorithms for shortest paths problems in structured instances, 2014.

[46] Duc-Cuong Dang, Aneta Neumann, Frank Neumann, Andre Opris, and Dirk Sudholt. Theoretical analysis of quality diversity algorithms for a classical path planning problem, 2024.

[47] Maximilian Probst Gutenberg and Christian Wulff-Nilsen. Fully-dynamic all-pairs shortest paths: Improved worst-case time and space bounds, 2020.

[48] Julia Chuzhoy and Ruimin Zhang. A new deterministic algorithm for fully dynamic all-pairs shortest paths, 2023.

[49] Hong Duc Bui, Shashwat Chandra, Yi-Jun Chang, Michal Dory, and Dean Leitersdorf. Improved all-pairs approximate shortest paths in congested clique, 2024.

[50] Xiao Mao. Fully-dynamic all-pairs shortest paths: Likely optimal worst-case update time, 2024.

[51] Udit Agarwal and Vijaya Ramachandran. A deterministic distributed algorithm for weighted all pairs shortest paths through pipelining, 2018.

[52] ChengSheng Wang, Xudong Wu, and Penghui Yao. Complexity of eccentricities and all-pairs shortest paths in the quantum congest model, 2022.

[53] Virginia Vassilevska Williams and Yinzhan Xu. Truly subcubic min-plus product for less structured matrices, with applications, 2019.

[54] Mahdi Boroujeni, Sina Dehghani, Soheil Ehsani, MohammadTaghi HajiAghayi, and Saeed Seddighin. Subcubic equivalences between graph centrality measures and complementary problems, 2019.

[55] Lijie Chen and Ryan Williams. An equivalence class for orthogonal vectors, 2018.

[56] Kasper Green Larsen and Ryan Williams. Faster online matrix-vector multiplication, 2016.

[57] Jesper Nederlof and Karol Węgrzycki. Improving schroeppel and shamir's algorithm for subset sum via orthogonal vectors, 2021.

[58] Karl Bringmann and Sebastian Krinninger. A note on hardness of diameter approximation, 2018.

[59] Bertie Ancona, Monika Henzinger, Liam Roditty, Virginia Vassilevska Williams, and Nicole Wein. Algorithms and hardness for diameter in dynamic graphs, 2019.

[60] Amir Abboud, Virginia Vassilevska Williams, and Joshua Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter, 2015.

[61] Raphael Yuster. Computing the diameter polynomially faster than apsp, 2011.

[62] Holger Dell and John Lapinskas. Fine-grained reductions from approximate counting to decision, 2020.

[63] Thiago Bergamaschi, Monika Henzinger, Maximilian Probst Gutenberg, Virginia Vassilevska Williams, and Nicole Wein. New techniques and fine-grained hardness for dynamic near-additive spanners, 2021.

[64] Peyman Afshani. A lower bound for dynamic fractional cascading, 2020.

[65] Holger Dell, John Lapinskas, and Kitty Meeks. Nearly optimal independence oracle algorithms for edge estimation in hypergraphs, 2024.

[66] Amir Abboud, Shon Feller, and Oren Weimann. On the fine-grained complexity of parity problems, 2021.

[67] Hodaya Barr, Tsvi Kopelowitz, Ely Porat, and Liam Roditty. $\{-1, 0, 1\}$-apsp and (min,max)-product problems, 2019.

[68] Danupon Nanongkai. Distributed approximation algorithms for weighted shortest paths, 2014.

[69] Michal Dory and Merav Parter. Exponentially faster shortest paths in the congested clique, 2020.

[70] Elli Anastasiadi, Antonis Antonopoulos, Aris Pagourtzis, and Stavros Petsalakis. Parameterized fine-grained reductions, 2019.

[71] David Coudert, Guillaume Ducoffe, and Alexandru Popa. Fully polynomial fpt algorithms for some classes of bounded clique-width graphs, 2017.

[72] Shiri Chechik, Yang P. Liu, Omer Rotem, and Aaron Sidford. Constant girth approximation for directed graphs in subquadratic time, 2020.

[73] Udit Agarwal, Vijaya Ramachandran, Valerie King, and Matteo Pontecorvi. A deterministic distributed algorithm for exact weighted all-pairs shortest paths in $\tilde{O}(n^{3/2})$ rounds, 2018.

[74] Udit Agarwal and Vijaya Ramachandran. Faster deterministic all pairs shortest paths in congest model, 2020.