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

The interconnected concepts in theoretical computer science, including SETH, 3SUM, and APSP, significantly enhance our comprehension of computational complexity and efficient computation limits. These concepts form a network that contributes to establishing lower bounds for various computational problems. For example, the 3SUM conjecture is pivotal in deriving conditional lower bounds across numerous problems [1]. Recognizing these interconnections is essential for modeling complex systems, as illustrated by the theoretical frameworks in [2].

Moreover, the pursuit of universally optimal algorithms, as examined in [3] and [4], highlights the practical implications of these interconnected concepts, particularly in distributed networks where efficient optimization is critical. The choice of appropriate distance closures in complex network analysis further illustrates how these concepts can deepen our theoretical insights [5].

The demand for breakthroughs in optimization algorithms, discussed in [6], underscores the dynamic nature of these interconnected ideas. Current algorithms often fail to adapt to real-time conditions, necessitating a more profound understanding of the underlying complexities. Thus, exploring these interconnected concepts is vital for advancing computational feasibility and developing innovative solutions to complex theoretical problems.

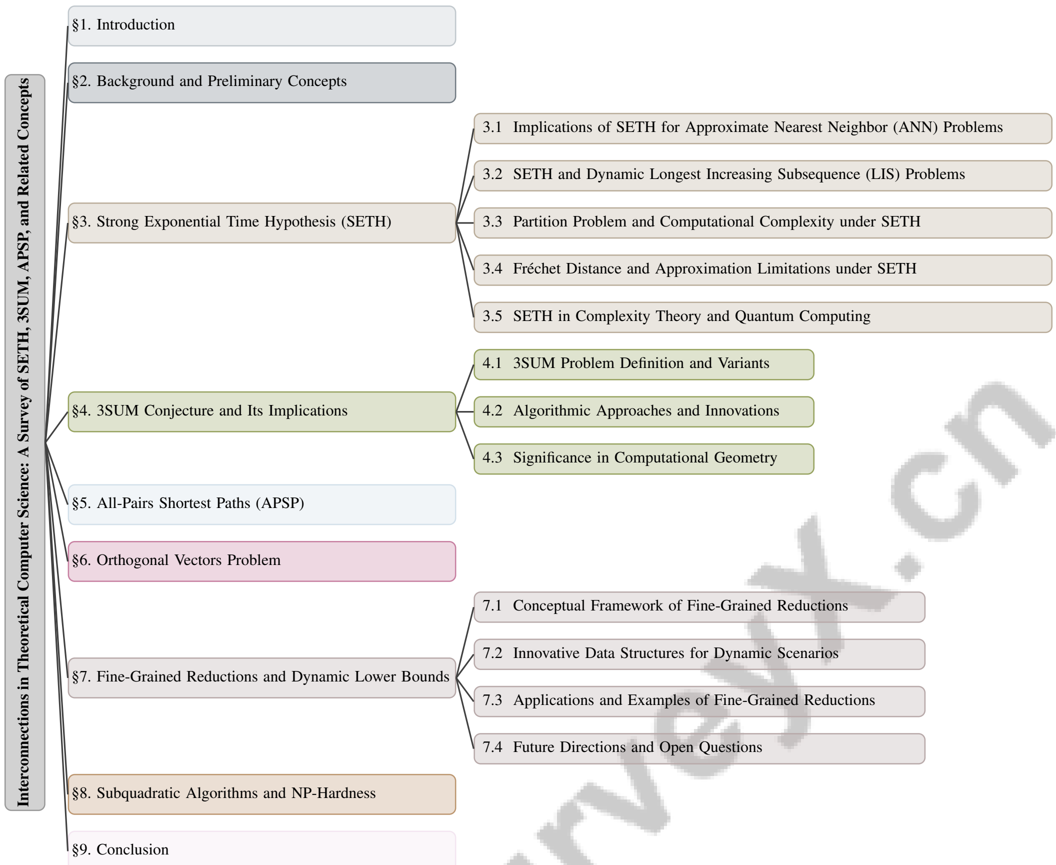

Figure 1: chapter structure

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 influence on computational complexity.

Subsequent sections provide background and foundational concepts, setting the stage for a deeper exploration of the theoretical principles that underpin research in computational complexity.

The survey then analyzes individual concepts, beginning with the Strong Exponential Time Hypothesis (SETH). This section discusses SETH's implications for SAT problems and its role in establishing computational lower bounds, including its relevance to Approximate Nearest Neighbor (ANN) and dynamic LIS, as well as its significance in quantum computing.

Following this, the 3SUM conjecture is examined, highlighting its critical role in computational geometry and complexity theory. The analysis of established reductions and conditional lower bounds demonstrates the conjecture's substantial impact on various computational problems, including dynamic data structures and graph algorithms. Notably, refinements to initial reductions proposed by Pătraşcu have led to more efficient connections between the 3SUM problem and offline SetDisjointness and SetIntersection, resulting in polynomially higher lower bounds for algorithms such as triangle enumeration in graphs and maintaining dynamic longest increasing subsequences. These advancements underscore the conjecture's importance in establishing optimality results and new conditional lower bounds across diverse contexts, including dynamic graph problems and pattern matching [7, 8].

The section on All-Pairs Shortest Paths (APSP) explores its computational complexity, known algorithms, and connections to SETH and 3SUM, addressing recent advancements and open questions in the field.

AI-generated, for reference only.

Next, the Orthogonal Vectors Problem is discussed, focusing on its relevance in fine-grained complexity and connections to other key problems, while also examining algorithmic approaches and potential applications.

The survey further investigates fine-grained reductions and dynamic lower bounds, elucidating their role in understanding the limits of efficient computation. This section highlights innovative data structures and proposes future research directions.

The concluding section addresses subquadratic algorithms and NP-hardness, discussing the challenges posed by NP-hard problems and the significance of efficient algorithm discovery. It also considers dynamic and fine-grained complexity in the quest for subquadratic solutions.

The synthesis of insights from the survey emphasizes the critical interconnections among complex network concepts, particularly the influence of distance closures on understanding indirect associations and network structures. It outlines promising research avenues, such as exploring alternative distance metrics and their implications for community detection and algorithm efficiency in large-scale data processing frameworks like MapReduce. This reflection underscores the need to consider topological distortions in complex network analyses and the potential for methodological advancements in both theoretical and applied contexts [9, 5, 10, 1].The following sections are organized as shown in Figure 1.

2 Background and Preliminary Concepts

2.1 Theoretical Frameworks in Computational Complexity

Theoretical frameworks in computational complexity, such as fine-grained complexity, provide a foundation for understanding the limits of efficient computation and establishing precise lower bounds for significant problems. These frameworks are particularly relevant in distributed settings, where they guide the development of universally optimal algorithms by considering graph parameters [4]. Addressing large datasets requires innovative approaches like the minimum degree algorithm, which mitigates the cost of solving sparse linear equations [11]. Massively parallel algorithms further tackle large-scale graph problems by optimizing communication and memory [12].

Graph representations are crucial in fields like molecular engineering, where accurate factorization of weighted graphs is necessary [13]. The study of distance closures in complex networks reveals the theoretical interplay, establishing isomorphisms between fuzzy and distance graphs [5]. In dynamic graphs, traditional methods with polynomial update times face limitations, necessitating advancements in data structures and sensitivity oracles for efficient shortest path computation amid edge failures [14, 15].

Quantum complexity theory presents challenges in high-dimensional spaces, where classical algorithms struggle. Although quantum algorithms show potential, surpassing brute force in these dimensions remains difficult [16]. The rank-finding conjecture adds depth to the fine-grained hardness of approximation problems [17]. Efforts to reduce space complexity have made larger problem instances, like the Subset Sum problem, more feasible [18]. The dynamic k-mismatch problem, which calculates Hamming distances efficiently, exemplifies the need for effective computational methods [19].

The Balanced Separator problem addresses complex partitioning in low conductance graphs, requiring robust frameworks [20]. Additionally, the difficulty of finding multiple kings in tournaments, demanding (n^2) queries even with randomized algorithms, highlights the inherent complexity of certain graph problems [21]. These frameworks collectively advance computational complexity studies, fostering innovation in algorithm design and addressing complex challenges in graph algorithms and quantum computing. By leveraging well-connected graphs and introducing tools like the Distanced Matching game, researchers enhance algorithm efficiency for critical issues, including dynamic APSP and cut problems, while tackling discrete Ricci curvatures [22, 8, 9, 11, 23].

2.2 Energy Efficiency in Computational Models

Energy efficiency is a pivotal consideration in computational models, especially within distributed systems and large-scale computations. Theoretical computer science increasingly focuses on optimizing energy usage, as demonstrated by low-energy deterministic distributed algorithms that allow nodes

AI-generated, for reference only.

to remain inactive during non-essential computations, enhancing the scalability and efficiency of networks [6]. Challenges from super-linear complexities in graph problems, solvable in polynomial time but requiring significant communication in the CONGEST model, underscore the need for energy-efficient solutions [24].

Addressing these challenges involves frameworks that optimize communication and computation to minimize energy consumption. In dynamic database environments, efficient query evaluation amidst updates and constraints has led to new approaches for computational efficiency [25]. Traditional methods, often reliant on rerunning static algorithms or recalculating distances after updates, prove inefficient, particularly in dynamic scenarios like the dynamic k-mismatch problem [19].

Integrating frameworks such as the multiplicative-weights framework with shortest-path computations simplifies processes like finding balanced sparse cuts, contributing to energy-efficient computational models [20]. These frameworks not only enhance computational efficiency but also play a vital role in advancing theoretical computer science by addressing energy constraints in large-scale and distributed computations.

In recent years, the Strong Exponential Time Hypothesis (SETH) has garnered significant attention within the field of computational complexity theory. Its implications extend across a variety of computational problems, necessitating a closer examination of its effects on algorithmic performance. Figure 2 illustrates the impact of SETH on several key areas, including Approximate Nearest Neighbor (ANN) problems, dynamic Longest Increasing Subsequence (LIS) problems, the partition problem, and Fréchet distance approximations. Each section of the figure elucidates the structural connections, challenges, and constraints imposed by SETH, thereby emphasizing the theoretical limits on algorithmic efficiency. Furthermore, it underscores the pressing need for innovative strategies to navigate these bounds, particularly in the realms of complexity theory and quantum computing. This comprehensive overview not only enhances our understanding of SETH but also highlights the critical interplay between theoretical constructs and practical algorithmic design.

3 Strong Exponential Time Hypothesis (SETH)

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

The Strong Exponential Time Hypothesis (SETH) profoundly affects the complexity of Approximate Nearest Neighbor (ANN) problems, highlighting their intrinsic hardness. The connection between ANN and the Shortest Vector Problem (SVP) implies that if SETH holds, solving ANN mirrors the challenges of SVP due to their structural similarities [26]. This link extends SETH's implications into computational geometry, associating ANN complexities with classical conjectures like kSUM [27]. For instance, the c-Approximate Near Neighbors problem (c-ANN), which involves preprocessing curves to determine proximity within a specified Fréchet distance, faces substantial challenges in constructing efficient data structures, with SETH suggesting sub-exponential time complexity is unlikely [28]. Quantum algorithms, despite their potential, show limited benefits for ANN, reinforcing SETH's constraints on computational efficiency [29]. This aligns with the idea that data structures like strings can leverage repetitive patterns for lower running times, yet remain bounded by SETH's theoretical limits [30].

3.2 SETH and Dynamic Longest Increasing Subsequence (LIS) Problems

SETH significantly influences the complexity of dynamic Longest Increasing Subsequence (LIS) problems, particularly regarding efficient LIS maintenance under dynamic conditions. The dynamic LIS problem, which involves updating the LIS as elements are added or removed, presents considerable computational challenges. Gawrychowski et al. [8] provide conditional lower bounds illustrating the difficulty of efficient updates in dynamic LIS scenarios. SETH suggests that problems like SAT cannot be solved in sub-exponential time, affecting dynamic LIS complexity. These bounds indicate that no algorithm can achieve sub-linear update times, specifically $O(n^{1/2-\epsilon})$ or $O(n^{1/3-\epsilon})$, for any constant $\epsilon > 0$. This underscores the broader implications for computational efficiency in dynamic data structures and the limitations imposed by conjectures like All-Pairs Shortest Paths and 3SUM [31, 1, 29, 7, 8]. The complexity of dynamic LIS is further complicated by efficiently handling arbitrary insertions and deletions, with significant improvements in update times necessitating breakthroughs challenging current SETH understanding [8].

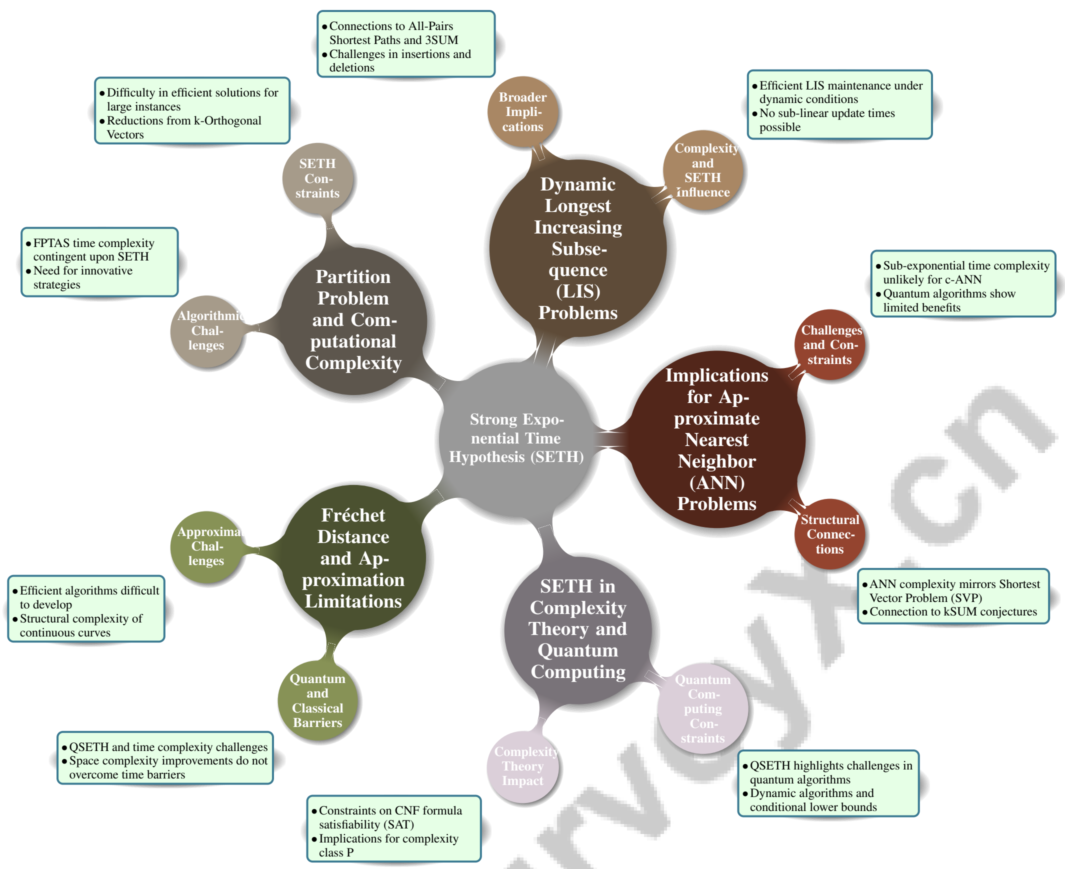

Figure 2: This figure illustrates the impact of the Strong Exponential Time Hypothesis (SETH) across various computational problems, highlighting its implications for Approximate Nearest Neighbor (ANN) problems, dynamic Longest Increasing Subsequence (LIS) problems, the partition problem, Fréchet distance approximations, and its role in complexity theory and quantum computing. Each section details the structural connections, challenges, and constraints imposed by SETH, emphasizing the theoretical limits on algorithmic efficiency and the need for innovative strategies to navigate these bounds.

3.3 Partition Problem and Computational Complexity under SETH

SETH profoundly impacts the computational complexity of the partition problem, a fundamental challenge in dividing a set into two subsets with equal sums. Under SETH, achieving efficient solutions for large instances becomes difficult, as existing methodologies often fall short due to the hypothesis's constraints [32]. This complexity is exacerbated by reductions from other hard problems, such as k-Orthogonal Vectors, establishing SETH-hardness and underscoring the problem's computational intensity [33]. Conditional lower bounds from the All-Pairs Shortest Paths (APSP) conjecture indicate that no algorithm can support queries and updates in time $O(n^{1/2})$ and $O(n^{1/3})$, further illustrating SETH's constraints [8]. Recent research highlights theoretical obstacles in developing efficient algorithms for the partition problem, with the best-known Fully Polynomial-Time Approximation Scheme (FPTAS) operating in $O(n + 1/)$ time, contingent upon SETH, emphasizing the problem's inherent difficulty [1, 32, 10, 19]. This underscores the need for innovative strategies to navigate established lower bounds within the current theoretical framework.

3.4 Fréchet Distance and Approximation Limitations under SETH

The Fréchet distance, essential for curve comparison, faces significant approximation challenges under SETH. Determining the Fréchet distance between curves is influenced by SETH, suggesting that efficient approximation algorithms are inherently difficult to develop [34]. This difficulty is

AI-generated, for reference only.

compounded by the structural complexity of continuous curves, complicating algorithm development for effective approximate queries [28]. SETH's implications extend to quantum computation through the Quantum Strong Exponential-Time Hypothesis (QSETH), asserting that even quantum algorithms face substantial time complexity challenges for problems like Edit Distance, requiring $\Omega(n^{1.5})$ time [29]. This indicates that both classical and quantum paradigms encounter barriers in efficiently solving curve similarity problems like the Fréchet distance. Advancements in problems like Subset Sum show potential for improving space complexity, yet these improvements do not overcome SETH's time complexity barriers for the Fréchet distance problem [18].

3.5 SETH in Complexity Theory and Quantum Computing

SETH is pivotal in shaping complexity theory, particularly in quantum computing contexts. It asserts that problems related to CNF formula satisfiability (SAT) cannot be solved more efficiently than exhaustive search, establishing fundamental constraints on computational efficiency. This conjecture significantly impacts complexity theory, underpinning lower bounds for various problems within complexity class P, indicating that algorithmic efficiency advancements are limited unless SETH is proven false [24, 11, 29, 7]. SETH establishes conditional lower bounds for computational problems, linking the complexity of approximating the Fréchet distance to SETH, suggesting challenges in developing efficient approximation algorithms [34]. This limitation extends to geometric problems like the Approximate Nearest Neighbor (ANN) problem, where achieving efficient algorithms under the norm remains unlikely without breakthroughs in understanding underlying hardness assumptions [26]. SETH's implications extend to quantum computing, where QSETH highlights constraints faced by quantum algorithms in solving problems like Edit Distance, requiring substantial time complexity [29]. This underscores the challenge of achieving quantum speed-ups for computational problems, as explored in the 3SUM problem context [35]. Despite quantum computing's potential, SETH and its quantum counterpart suggest certain problems may resist efficient quantum solutions. Exploring dynamic algorithms further illustrates SETH's impact on complexity theory, with conditional lower bounds for dynamic problems like All-Pairs Shortest Paths (APSP) and Orthogonal Vectors (OMv) emphasizing the difficulty of achieving efficient updates and queries [8]. The introduction of APSP solvers in distributed systems, like Apache Spark, demonstrates potential for leveraging distributed processing to address these challenges, albeit within SETH's constraints [36].

4 3SUM Conjecture and Its Implications

The 3SUM conjecture stands as a cornerstone in computational theory, elucidating the complexities inherent in this fundamental problem. It serves as a benchmark for understanding computational challenges across domains such as computational geometry, triangle enumeration, and dynamic graph algorithms. Recent advancements, including subquadratic algorithms, challenge the traditional $O(n^2)$ time complexity, reshaping our comprehension of the problem's core aspects and its diverse applications [1, 23].

4.1 3SUM Problem Definition and Variants

At its core, the 3SUM problem involves identifying three numbers a, b, and c from a set of n real numbers such that their sum is zero [23]. This problem is crucial for establishing conditional lower bounds in numerous domains [1]. The conjecture suggests a $\Theta(n^2)$ time requirement, influencing lower bounds for dynamic data structures and graph problems [7]. The Quantum-3SUM-Conjecture further posits the nonexistence of sublinear quantum algorithms for this problem [35].

Variants of 3SUM explore different computational contexts, such as detecting Abelian and additive square factors in strings [37], and maintaining solutions amid updates in dynamic graph scenarios [38]. The conjecture also intersects with computational geometry, influencing algorithms for approximate nearest neighbors under specific norms [26], and highlights computational challenges in distributed environments like MapReduce [10].

4.2 Algorithmic Approaches and Innovations

Innovations in algorithms for the 3SUM problem have significantly advanced efficiency, surpassing the $O(n^2)$ time complexity. A decision tree of depth $O(n^{3/2} \log n)$ and subquadratic algorithms

Method Name	Algorithmic Efficiency	Problem Reductions	Computational Implications
3SUM[23]	Subquadratic Algorithms	Decision Tree Approach	Computational Geometry
ER3SUM[7]	Time Complexity Improvements	Efficient Reductions	Conditional Lower Bounds
MR-APSP-Matrix-3SUM[10]	Constant Round Algorithms	Problem Reductions	Broader Impact
QSG3SUM[39]	Quadratic Speedup	Search Problems	Broader Applications

Table 1: This table presents a comparative analysis of various algorithmic methods for addressing the 3SUM problem, highlighting their algorithmic efficiency, problem reduction strategies, and computational implications. The methods include advancements such as subquadratic algorithms, time complexity improvements, constant round algorithms, and quadratic speedup, each contributing to the broader impact on computational geometry and related fields.

exemplify these advancements, employing strategies like input sorting and group partitioning for efficient searches [23]. Demonstrating 3SUM-hardness extends to problems like detecting odd half-length Abelian squares and computing centers of Abelian square factors [37]. Table 1 provides a comprehensive overview of innovative algorithmic approaches to the 3SUM problem, illustrating the advancements in efficiency and problem-solving strategies that have implications for computational geometry and beyond.

New reductions from 3SUM to offline SetDisjointness and SetIntersection illustrate innovative approaches to leveraging 3SUM's complexity [7]. In distributed computing, novel MapReduce algorithms optimize resource utilization, enhancing scalability and performance for fundamental problems like 3SUM [10]. Quantum algorithms also offer promising avenues, generalizing techniques to address geometric problems beyond traditional solutions [39].

Conditional hardness results for problems like $(1+\epsilon)$-approximate nearest neighbor under ℓ_∞ norm underscore 3SUM's broader implications in computational geometry, where efficient algorithms hinge on hypotheses such as SETH [26]. Recent refutations of the 3SUM conjecture's optimality have led to subquadratic algorithms impacting challenges like k-variate linear degeneracy testing and dynamic graph algorithms [1, 7, 23].

4.3 Significance in Computational Geometry

The 3SUM problem's significance in computational geometry is highlighted by its role in establishing conditional lower bounds for geometric challenges. Its conjecture aids in understanding the complexity of geometric configurations and solving related problems efficiently [1]. The identification of 3SUM-hardness in detecting Abelian and additive square factors underscores its intricate connections to geometric problem-solving [37].

Quantum speedups for geometric 3SUM-hard problems emphasize 3SUM's importance in computational geometry. Despite quantum computing's potential to overcome classical limitations, the classical 3SUM problem remains central to understanding quantum speedup limits [35]. The need for parallel approaches in problems like All-Pairs Shortest Paths (APSP) is accentuated by practical applications' increasing graph sizes [36].

Algorithmic advancements, such as those by Grønlund and Pettie, demonstrate significant improvements over prior methods, challenging the conjectured optimal time complexity [23]. These innovations enhance our understanding of 3SUM, opening avenues for future research and potential applications to other geometric and algorithmic complexity problems.

5 All-Pairs Shortest Paths (APSP)

5.1 Dynamic and Decremental APSP

Dynamic and decremental APSP algorithms are pivotal for maintaining shortest paths in graphs with frequent edge changes. Recent innovations have introduced algorithms that maintain nearly optimal spanners during edge updates, achieving total processing times of $m^{1+o(1)}$ with sub-linear recourse, particularly when insertions exceed deletions. By integrating a multiplicative-weights framework with Dijkstra's algorithm, efficient identification of balanced sparse cuts is achieved with almost-linear time complexity, simplifying previous approaches. A novel dynamic APSP algorithm provides efficient query access to approximate distances while processing edge updates in $|E|^{1+o(1)}$ time, leveraging dynamic spanners for optimized performance without complex expander structures [40, 20, 38].

Advancements in decremental APSP algorithms offer improved approximation guarantees for weighted and unweighted graphs, crucial for managing decremental changes in large networks [41, 42]. Distributed computing has further enhanced dynamic APSP, with Huang's method achieving exact weighted APSP in $\tilde{O}(n^{5/4})$ rounds, showcasing distributed algorithms' potential to improve scalability and efficiency [43]. Agarwal et al. introduce deterministic techniques for constructing successive blocker sets, enabling shortest cycle computation in $\tilde{O}(n)$ rounds, illustrating deterministic strategies' effectiveness in optimizing dynamic graph algorithms [44].

Fast matrix multiplication and bridging sets have been employed to enhance APSP efficiency [45]. Quantum computing, combining classical matrix multiplication with quantum search, has yielded faster APSP computation times, especially in structured graphs [46]. These advancements tackle computational challenges in dynamic graph scenarios, enhancing APSP algorithms' applicability across various domains. Innovations like the Distanced Matching game for well-connected graphs improve deterministic algorithms' performance in distributed settings, achieving bounds as low as $\tilde{O}(n^{3/2})$ rounds for weighted APSP [9, 47, 48].

5.2 Distributed and Parallel Approaches

Distributed and parallel methodologies for APSP are essential for large-scale graphs and dynamic networks. Traditional algorithms like Floyd-Warshall and Johnson are impractical for large graphs, prompting parallel approaches. Recent advancements with Apache Spark have solved APSP problems with over 200,000 vertices on a 1024-core cluster, highlighting the importance of auxiliary shared persistent storage to overcome Spark's limitations. Research into fully dynamic APSP algorithms aims to optimize update times while maintaining efficient space utilization, with the best-known achieving a worst-case running time of $\tilde{O}(n^{2.5})$. A novel deterministic distributed algorithm computes exact weighted APSP in $\tilde{O}(n^{3/2})$ rounds, significantly improving over previous methods [49, 48, 36]. These methodologies leverage graph computations' inherent parallelism to optimize computational and communication overheads, enhancing scalability and performance.

Huang's method integrates distributed algorithms to compute shortest paths efficiently, managing edge weights and communication constraints [43]. Theoretical analysis by Dang evaluates distributed algorithms' efficiency by measuring fitness evaluations required for computing shortest paths, highlighting their effectiveness in reducing computational complexity [50]. Evald introduces innovative data structures for decremental APSP in directed graphs, enhancing robustness and efficiency, particularly in dynamic scenarios [42]. Parallel computing techniques, as demonstrated by Zwick, involve computing distance products of matrices representing the graph, significantly boosting computational throughput [45].

Future research could explore developing deterministic algorithms achieving the current randomized bound of $\tilde{O}(n)$ for APSP. This includes innovative algorithmic strategies leveraging well-connected graphs and the Distanced Matching game, optimizing existing methods to enhance performance in static and dynamic scenarios [9, 20, 48]. This pursuit aims to yield more efficient and scalable solutions in distributed and parallel APSP algorithms.

5.3 Approximation and Quantum Algorithms

Exploration of approximation and quantum algorithms for APSP is crucial for improving computational efficiency in large-scale and dynamic graphs. Recent advancements include new approximate decremental APSP algorithms that reduce update times, achieving (2 +)-approximation with total update times as low as $\tilde{O}(m^{1/2}n^{3/2})$ for weighted graphs with edge deletions, alongside fully dynamic algorithms maintaining distance matrices under vertex insertions and deletions with enhanced space efficiency [49, 51, 41, 52]. These innovations address complexities in dynamic graph structures and establish new benchmarks for approximation accuracy and speed.

Dory et al. introduce decremental APSP algorithms that outperform existing methods in update times and approximation ratios, offering effective solutions for maintaining shortest paths amid edge deletions [41]. Distributed computing techniques have advanced APSP solutions, with Huang proposing a distributed algorithm efficiently computing exact weighted shortest paths while optimizing communication rounds [43]. In quantum algorithms, Nayebi et al. 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 accelerate APSP computations [46]. However, quantum algorithms may face fundamental complexity limits, as Wang et al. indicate no quantum speedup for computing eccentricities and APSP [53].

The development of efficient algorithms for the Min-Plus product achieves the first truly subcubic algorithm for less structured matrices, enhancing computational efficiency for related problems, including APSP [54]. Complementary versions of graph centrality problems, such as CoDiameter, CoMedian, and CoRadius, demonstrate subcubic equivalences with APSP, providing insights into APSP-related challenges' complexity [55]. These innovations underscore APSP's interconnected nature with broader computational challenges, driving further 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 fundamental challenge in theoretical computer science, focusing on identifying orthogonal pairs among n binary vectors in $\{0,1\}^{O(\log n)}$ [56]. It plays a crucial role in understanding fine-grained complexity and is foundational to problems like 3SUM and APSP [10]. Existing algorithms require $O(n^2 \log n)$ time, with conditional lower bounds suggesting that sub-quadratic solutions, such as $O(n^{1.99})$, are unlikely [57, 56, 58]. The exponential growth of the vector space with increasing dimensionality complicates traditional approaches. Recent advancements have led to constant-round algorithms that enhance efficiency, particularly in large-scale data processing environments, by leveraging parallelism in distributed computing [10]. The OV problem also impacts network analysis, especially in estimating network diameter, essential for understanding node distances [59]. Its interrelations with classical problems like Subset Sum highlight the interconnected nature of computational challenges in theoretical computer science.

6.2 Reductions and Algorithmic Approaches

The OV problem is central to fine-grained complexity, offering insights into the computational limits of related problems. Recent reduction frameworks enhance the understanding of its interrelationships with various computational challenges [56]. These frameworks establish conditional lower bounds by using OV complexity as a benchmark. Reduction techniques often transform problems into high-dimensional vector spaces to identify orthogonal pairs, crucial for developing algorithms that manage exponential vector dimension growth. The introduction of constant-round algorithms for OV exemplifies potential computational overhead optimization in large-scale environments [10]. OV's relation to computing or approximating diameters in sparse graphs, where $\tilde{\Omega}(n)$ rounds are necessary, emphasizes its broader implications in network analysis [59]. Algorithmic approaches increasingly focus on distributed and parallel computing to improve efficiency, leveraging frameworks like MapReduce, Hadoop, and Spark to optimize memory and machine utilization. By balancing these factors, algorithms minimize computational rounds for complex tasks such as Matrix Multiplication and APSP, fully exploiting distributed environments [4, 10]. These innovations are vital for advancing theoretical computer science, offering scalable solutions to foundational problems like OV.

6.3 Applications and Implications

The OV problem has significant applications across theoretical computer science, serving as a benchmark for understanding computational complexity in high-dimensional data challenges. A primary application is establishing conditional lower bounds for computational tasks, enabling derivation of hardness results for problems like APSP and diameter estimation in sparse graphs, which require substantial computational resources [59]. In network analysis, OV is crucial for approximating characteristics like diameter, vital for evaluating network performance and connectivity. Identifying orthogonal pairs among vectors parallels evaluating maximum node distances, both essential for understanding network efficiency and reliability. The computational characteristics of OV align with network analysis challenges, including APSP, highlighting the importance of closure choices in network distance calculations [5, 56, 58]. The implications of OV extend into quantum computing, where adapting classical reductions to quantum contexts offers new insights into computational complexity [29]. This cross-disciplinary approach suggests potential for quantum algorithms to tackle OV challenges, though significant barriers to substantial quantum speedups remain. Exploring OV informs algorithmic development in classical problems like Subset Sum, providing a framework for understanding efficient computation limits in high-dimensional spaces. The intersections among these problems underscore the interconnected nature of computational challenges, where progress in one area can drive breakthroughs in another.

7 Fine-Grained Reductions and Dynamic Lower Bounds

Fine-grained reductions are critical in computational complexity, highlighting nuanced relationships among problems and maintaining computational hardness while exploring lower bounds across diverse challenges. This section explores how these reductions advance complexity theory and foster innovative algorithmic solutions, setting the stage for a detailed analysis of their implications and applications.

7.1 Conceptual Framework of Fine-Grained Reductions

The framework of fine-grained reductions is central to understanding the intricate relationships among computational problems. It utilizes reductions to transform one problem into another, preserving computational hardness and establishing conditional lower bounds across domains. For example, reductions from the 3SUM problem to those involving Abelian and additive squares underscore the complexity of these problems through combinatorial techniques [37]. Fine-grained reductions are also effective in approximating counting problems, employing decision oracles to lower computational complexity [60].

This framework extends to dynamic algorithms, such as maintaining a (1 + ,)-spanner on undirected graphs during edge updates, demonstrating adaptability in evolving data structures [61]. It also examines the limitations of dynamic fractional cascading in the pointer machine model, offering insights into constraints of dynamic computational models [31]. The use of Boolean matrix structures and the Orthogonal Vectors problem illustrates the versatility of fine-grained reductions in achieving faster computations [58]. Applications from planar graphs to polynomial lower bounds for dynamic LIS problems exemplify innovative uses of these reductions [8].

The framework's depth is further highlighted by the relationship between approximate counting and decision problems in hypergraphs [62]. In dynamic graph scenarios, maintaining vertex sparsifiers with subpolynomial update times showcases practical applications of fine-grained reductions in optimizing dynamic algorithms [14]. Additionally, space-reducing algorithms, like those for the Subset Sum problem achieving $O^*(2^{0.246n})$, demonstrate the potential of these reductions to enhance computational efficiency [18]. Trade-off algorithms in dynamic k-mismatch problems further illustrate innovative solutions enabled by fine-grained reductions [19].

Recent advancements include new upper bounds that allow better approximation factors with linear preprocessing time [28]. Collectively, these developments underscore the critical role of fine-grained reductions in advancing theoretical computer science and exploring computation limits.

7.2 Innovative Data Structures for Dynamic Scenarios

Innovative data structures are crucial for addressing dynamic graph environments where data frequently changes. Recent advancements have introduced dynamic approximation algorithms leveraging randomized techniques for efficient updates, maintaining accurate graph parameter estimates [63]. A notable innovation is the DynamicSpanner algorithm, which maintains low congestion in embedding paths, preserving spanner properties during updates, essential for large-scale networks [40].

These advancements highlight the importance of designing adaptable algorithms that minimize computational costs. By employing randomized techniques and ensuring low edge congestion, new data structures provide efficient solutions for dynamic graph problems like APSP and SSSP amid edge updates. These improvements not only enhance theoretical frameworks but also have practical implications, including flow-routing algorithms previously reliant on randomized methods. Vertex sparsifiers preserving terminal node distances further strengthen these solutions, enabling efficient updates in dynamic environments [9, 14].

7.3 Applications and Examples of Fine-Grained Reductions

Fine-grained reductions are pivotal in theoretical computer science, establishing precise connections between computational problem complexities. They aid in understanding problem difficulty and facilitate efficient algorithm development. A notable application is in classifying parity problems, detailing complexity relationships and providing insights into equivalence classes [64]. The Orthogonal Vectors problem is a cornerstone in fine-grained complexity, expanding equivalence classes with new tools for lower bounds [56]. Reductions from OV to diameter approximation problems demonstrate shared complexity, highlighting interconnected computational challenges [59].

In graph theory, fine-grained reductions impact problems involving matrix operations and the Min-Plus product. Developing truly subcubic algorithms for the Min-Plus product exemplifies their potential to enhance computational efficiency [54]. These advancements optimize algorithm performance in large-scale graph environments. The application extends to dynamic algorithms, employing techniques to assess performance by measuring update times and structure quality, such as spanners [61]. The introduction of space-efficient reductions in problems like Subset Sum, using refined algorithms for detecting orthogonal pairs, showcases innovative uses of fine-grained reductions to solve complex problems with limited resources [57].

7.4 Future Directions and Open Questions

Exploring fine-grained reductions offers opportunities for advancing computational complexity understanding, with several promising directions and open questions. One area involves optimizing algorithms for combinatorial optimization problems, particularly in partition contexts [32]. In dynamic graph problems, refining algorithms for resilience to edge deletions and exploring well-connected graphs remain critical [9]. Quantum advantages in weighted networks and distance parameters present intriguing research directions, including refining quantum protocols for efficiency in eccentricities and APSP problems [53]. Further optimizations of the target-(min,max)-product problem and exploring additional reductions to enhance APSP algorithm performance are promising areas [65].

Refining distributed approximation algorithms for different network topologies is another vital research area, potentially extending to other combinatorial optimization problems [66]. The Quantum Strong Exponential Time Hypothesis framework offers fertile ground for exploring computational problems, particularly in understanding compression-oblivious properties [29]. Further optimizations of reductions, such as those related to the 3SUM problem, remain open questions [7]. Exploring emulator construction processes and their application to other graph problems, including extensions to directed graphs, offers another promising research direction [67].

The exploration of fine-grained reductions reveals a vibrant landscape in computational complexity research, as highlighted by advancements like the parameterized fine-grained reductions framework, enhancing understanding of improvable problems and facilitating efficient algorithm development. This dynamic field addresses theoretical challenges while identifying practical applications, particularly in distributed processing environments like MapReduce, where innovative algorithms have been designed for fundamental problems like Matrix Multiplication and APSP. Collectively, these future directions and unresolved questions present opportunities to deepen our insight into computational complexity and its real-world implications [10, 68].

8 Subquadratic Algorithms and NP-Hardness

8.1 Deterministic Approaches and Dynamic Graphs

Deterministic subquadratic algorithms play a vital role in dynamic graph analysis, particularly for decremental shortest paths and dynamic longest increasing subsequences. These algorithms offer efficient updates and queries by maintaining data structures that accommodate adaptive adversaries, surpassing traditional randomized methods that depend on oblivious adversaries. Recent advances have achieved optimal total update times for decremental single-source and all-pairs shortest paths, setting new performance standards [69, 70, 8]. The ongoing challenge is to develop algorithms that handle dynamic changes, such as edge modifications, while maintaining subquadratic time complexity.

A key advancement is the development of algorithms providing constant factor approximations for graph girth in subquadratic time, significant for network design and analysis [71]. Deterministic algorithms for dynamic graph problems leverage sophisticated data structures to ensure rapid updates and efficient queries, crucial for real-time applications that require adaptability without compromising speed or accuracy [51, 47, 70, 48, 19].

The exploration of deterministic methods is intertwined with fine-grained complexity research, which seeks to define computational boundaries. Efficient maintenance of shortest path information, spanners, and emulators is essential, especially for dense graphs. Conditional lower bounds based on fine-grained complexity hypotheses reveal inherent efficiency limits, such as those posed by the OMv conjecture on dynamic algorithm performance [61, 9, 63, 68]. By establishing subquadratic bounds, researchers enhance understanding of problem difficulty and push computational feasibility limits.

8.2 3SUM-Hardness and Related Problems

3SUM-hardness is central to computational complexity, defining the limits of efficient computation across various problems. The 3SUM conjecture suggests no strongly subquadratic algorithm exists for solving the 3SUM problem, impacting related problems by setting a computational difficulty benchmark [7]. Demonstrating 3SUM-hardness involves proving that certain problems cannot be solved in strongly subquadratic time unless the conjecture is false, as seen in problems like detecting Abelian and additive square factors [37].

The challenge of subquadratic solutions is pronounced in geometric problems, where the 3SUM conjecture limits algorithmic efficiency [39]. The conjecture also influences parameterized complexity, where fixed-parameter improvements aim to enhance efficiency within hardness constraints [68]. The Approximate Nearest Neighbor (ANN) problem exemplifies these challenges, with established hardness results indicating the need for breakthroughs to overcome 3SUM-imposed barriers [26].

The relevance of 3SUM-hardness extends to problems like k-Orthogonal Vectors and k-SUM, with conditional lower bounds highlighting the interconnected nature of computational challenges [62].

8.3 Graph Parameters and Subquadratic Time

Investigating graph parameters and their relation to subquadratic algorithms is crucial for enhancing computational efficiency. Identifying graph parameters that facilitate subquadratic algorithms would significantly improve graph-related computations. Parameters like radius and diameter, based on vertex eccentricities, are vital in applications, with current methods often requiring at least (n^2) time for n-node graphs. Recent research explores approximation and fixed-parameter algorithms for potentially subquadratic solutions, emphasizing feasibility and limitations [55, 11, 69, 24].

A notable contribution is the refined understanding of diameter approximation's relation to the Orthogonal Vectors problem, enhancing algorithm efficiency in distributed computing [59]. The study of Fréchet distance and its variants highlights complexities in developing subquadratic algorithms for graph problems [34].

Integrating fine-grained complexity techniques into graph parameter studies offers opportunities for subquadratic solutions. Establishing conditional lower bounds and exploring reductions between graph problems and computational challenges can reveal optimization opportunities. Recent studies show that maintaining the longest increasing subsequence (LIS) with updates in sub-linear time is infeasible under certain conjectures, while advancements in 3SUM reductions lead to higher lower bounds on dynamic data structures [11, 7, 8].

8.4 Dynamic and Fine-Grained Complexity

Exploring dynamic and fine-grained complexity is crucial for advancing subquadratic algorithms, as it reveals how problem structures affect computational efficiency. Parameterized fine-grained reductions have uncovered intricate relationships that transfer performance improvements across problems, leading to subquadratic fixed-parameter algorithms for challenges like the orthogonal vectors problem. Integrating fine-grained complexity with distributed processing frameworks, such as MapReduce, has enabled efficient algorithms for fundamental tasks, emphasizing trade-offs between memory usage and machine count. Optimal fine-grained hardness results for approximating linear systems solutions highlight critical connections between computational problems [17, 10, 68].

Dynamic complexity focuses on efficiently maintaining solutions as data changes, crucial for real-world applications. Fine-grained complexity establishes precise computational limits and nuances differentiating problems by inherent difficulty. Recent advancements improve approximation guarantees and methods for dynamic graphs, where efficient solutions amid changes are vital. Exploring parity versions of problems reveals additional complexity characteristics, enriching understanding of problem hardness [64].

Pursuing subquadratic algorithms is informed by techniques enhancing online computations, such as matrix-vector multiplication, foundational for dynamic problem research [58]. These techniques advance algorithmic design, particularly in dynamic settings where efficient update handling is crucial.

Parameterized fine-grained reductions (PFGR) offer a framework for understanding fixed-parameter improvements (FPI) and traditional parameterized complexity classes, presenting new avenues for extending these concepts to other computational problems [68]. However, challenges persist in generalizing methods to graphs with negative edge weights or specific topologies where blocker sets may not cover all paths [44]. Future research could focus on refining algorithms, exploring deterministic approaches, and extending methods to other NP-complete problems [57].

9 Conclusion

This survey underscores the intricate relationships among key concepts in theoretical computer science, such as SETH, 3SUM, and APSP, and their collective role in advancing our understanding of computational complexity. These interconnections are instrumental in establishing conditional lower bounds and elucidating the intrinsic challenges of computational problems. The exploration of parity problem variants reveals these can be equally or more complex than their original forms, as demonstrated by the Negative Weight Triangle problem. Fine-grained reductions have proven essential in complexity theory by transforming approximate counting problems into decision problems with minimal overhead.

The survey also highlights the critical role of energy efficiency in computational models, especially within distributed systems, where near-optimal performance can be achieved with reduced energy consumption. The development of universally optimal algorithms in hybrid models marks a significant step forward, showcasing the potential for optimal broadcasting in complex network environments.

Dynamic and fine-grained complexity insights reveal the adaptability of algorithms in changing contexts. The dynamic k-mismatch algorithm offers optimal trade-offs for updates and queries, surpassing existing methods. Efficient maintenance of graph parameters during dynamic changes is crucial for real-world application performance. The DynamicSpanner algorithm enhances previous approaches, offering robust solutions for edge-dynamic graphs. Future research is needed to optimize algorithms for weighted graphs and explore the effects of negative weights in mixed graphs.

The proposed data structures provide tight bounds for approximate nearest neighbor searching under the continuous Fréchet distance, surpassing previous methods and suggesting further exploration potential. Additionally, the equivalence of APSP and CoDiameter as subcubic problems, alongside similar equivalences for other centrality problems, advances the understanding 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] Tsvi Kopelowitz, Seth Pettie, and Ely Porat. Higher lower bounds from the 3sum conjecture, 2019.

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

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

[10] 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[48] 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.

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

[50] 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.

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

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

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

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

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

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

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

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

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

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

[61] 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.

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

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

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

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

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

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

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

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

[70] Julia Chuzhoy and Thatchaphol Saranurak. Deterministic algorithms for decremental shortest paths via layered core decomposition, 2020.

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

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.