國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Informational Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

三個導出子圖演算法——完美圖、最短奇洞、非最短路徑

Improved Algorithms for Recognizing Perfect Graphs and Finding Shortest Odd Holes and Non-shortest Induced Paths

邱允中

Yung-Chung Chiu

指導教授：呂學一 博士

Advisor: Hsueh-I Lu, Ph.D.

中華民國 111 年 7 月

July 2022

doi:10.6342/NTU202203481

Acknowledgement

首先非常非常非常感謝指導教授呂學一老師用心的指導。兩年前我完全是個演算法領域的門外漢，也絕對不是天賦最高的學生，但老師總是非常快速又耐心的回答我的每個非常基本的問題，也不吝惜花費大量的時間與我討論，有時直到深夜。印象深刻的是去年八月研究本篇論文第三個結果時，卡關在其中一個地方，不停的鬼打牆。老師就這樣用 messenger 陪我討論了一個小時，我非常感謝！在兩年前我加入 406 實驗室時，希望能夠磨練出自己未來需要的能力，而這兩年間，很感謝老師嚴格但實在的要求，我想我得到的比我曾經想像的更多：研究上大膽、進攻的假設、TOP-DOWN 的研究思維、如何給出一個有趣的 talk…我深信這些能力在研究之外、在我未來的人生中也將會帶給我很大的幫助。因此，對於能接受老師的指導，我感到非常幸運，再次感謝老師！也感謝開元學長在我對研究有所疑問時，提供我一些過來人的建議，獲益匪淺！

很感謝我的家人，我想能在一個可以專注於研究而不必為生活煩心的環境絕不該被視為理所當然。特別是疫情期間，長時間待在家中，在生活起居上受到父母深深的守護。也很感謝媽媽在去年夏天疫情期間，我處於低潮的時候，要我每天陪伴去公園散步，多了母子相處時間的同時，也舒緩了我的情緒，更常常帶給我研究上的靈感。

最後我想感謝我擁有的信仰，研究之路常常是孤獨、辛苦、更是充滿不確定性的，若內心不存在真實的信念，就不會有勇氣去超越自身的軟弱、怯懦、怠惰、逃避與絕望。因為有信仰，我才能夠確信自己正在往正確的方向前進，也能夠時時刻刻提醒自己讀研究所的原因，以及究竟想在這兩年內留下什麼。每當研究一籌莫展的時候，這樣的提醒總是能讓我又有力量走下去。也因為有這樣的信仰，我認識了一群能夠互相鼓勵的好夥伴！感謝台大大學會、內湖區還有和光弦樂團的大家。這兩年與大家的共戰絕對是我進行研究很大的動力。謝謝群越低潮時的電話以及談心。也謝謝搭檔圓融在我最近半年忙碌口試時互相 cover 大學部。特別想感謝思筠在我對研究感到徬徨時和我說「你一定可以的」，我深受鼓勵！

回想起這兩年充實的奮戰，我感到非常的幸運，能夠受到如此多人事物的守護，謝謝！

摘要

判斷一張圖中是否包含某種導出子圖，在圖論以及演算法領域的許多重要成果都扮演了重要角色。其中一個具有高度代表性的例子是「完美圖的辨識問題」，也就是判定一張圖是否符合「每個導出子圖中，最大團的頂點數都等於該導出子圖的著色數。」2006 年 Chudnovsky、Robertson、Seymour 以及 Thomas 證明了「強完美圖定理」這個獲得 2009 年 Fulkerson Prize 殊榮的成果，解決 Berge 在 1960 年留下的猜想，確認「一張圖是完美圖若且唯若它和它的補圖都不包含任何奇洞」，其中一張圖的奇洞是該圖的一個長度至少五且為奇數的導出環。根據這個定理，Chudnovsky、Cornuéjols、Liu、Seymour 以及 Vušković 提出了一個時間複雜度為 $O(n^9)$ 的演算法來辨認完美圖，其中 n 表示圖的節點數。在此篇論文中，我們改進了三個偵測/尋找一張有 n 個節點的圖 G 中的導出子圖的演算法。分別說明如下：

1. 是否存在一個多項式時間演算法來偵測「G 是否包含奇洞」在長達數十年期間一直是無人能解的懸案。Chudnovsky、Scott、Seymour 以及 Spirkl 終於在 *Journal of the ACM* 2020 年提出了第一個多項式時間偵測 G 中的奇洞的演算法，時間複雜度為 $O(n^9)$。Lai、Lu 以及 Thorup 之後在 *ACM STOC* 2020 改進到 $O(n^8)$，我們則進一步改進到 $O(n^7)$，從而也把辨識完美圖的的時間複雜度改進到 $O(n^7)$。
2. Chudnovsky、Scott 以及 Seymour 在 *ACM Transactions on Algorithms* 2021 年提出了第一個多項式時間尋找 G 中最短奇洞的演算法，時間複雜度為 $O(n^{14})$。我們將其改進到 $O(n^{13})$。
3. Berger、Seymour 以及 Spirkl 在 *Discrete Mathematics* 2021 年提出了第一個尋找 G 中給定兩點之間非最短導出路徑的演算法，時間複雜度為 $O(n^{18})$。我們將其大幅度改進到 $O(n^{4.75})$。這第三個成果已經發表在 *Proceedings of the 39th International Symposium on Theoretical Aspects of Computer Science* (STACS 2022)。

關鍵字：完美圖、導出子圖、奇洞、導出路徑、非最短路徑、動態資料結構

doi:10.6342/NTU202203481

Abstract

An *induced* subgraph of an n-vertex graph G is a graph that can be obtained by deleting a set of vertices together with its incident edges from G. A *hole* of G is an induced cycle of G with length at least four. A hole is *odd* (respectively, *even*) if its number of edges is odd (respectively, even). Various classes of induced subgraphs are involved in the deepest results of graph theory and graph algorithms. A prominent example concerns the *perfection* of G that the chromatic number of each induced subgraph H of G equals the clique number of H. The seminal Strong Perfect Graph Theorem proved in 2006 by Chudnovsky, Robertson, Seymour, and Thomas, conjectured by Berge in 1960, confirms that the perfection of G can be determined by detecting odd holes in G and its complement. Based on the theorem, Chudnovsky, Cornuéjols, Liu, Seymour, and Vušković show in 2005 an $O(n^9)$-time algorithm for recognizing perfect graphs, which can be implemented to run in $O(n^{6+\omega})$ time for the exponent $\omega < 2.373$ of square-matrix multiplication. We show the following improved algorithms for detecting or finding induced subgraphs in G.

1. The tractability of detecting odd holes in G was open for decades until the major breakthrough of Chudnovsky, Scott, Seymour, and Spirkl in 2020. Their $O(n^9)$-time algorithm is later implemented by Lai, Lu, and Thorup to run in $O(n^8)$ time, leading to the best formerly known algorithm for recognizing perfect graphs. Our first result is an $O(n^7)$-time algorithm for detecting odd holes, immediately implying a state-of-the-art $O(n^7)$-time algorithm for recognizing perfect graphs. Finding an odd hole based on Lai et al.'s $O(n^8)$-time algorithm for detecting odd holes takes $O(n^9)$ time.

doi:10.6342/NTU202203481

2. Chudnovsky, Scott, and Seymour extend in 2021 the $O(n^9)$-time algorithms for detecting odd holes (2020) and recognizing perfect graphs (2005) into the first polynomial-time algorithm for obtaining a shortest odd hole in G, which runs in $O(n^{14})$ time. Our second result is an $O(n^{13})$-time algorithm for finding a shortest odd hole in G.
3. For vertices u and v of an n-vertex graph G, a uv-*trail* of G is an induced uv-path of G that is not a shortest uv-path of G. In 2021, Berger, Seymour, and Spirkl gave the previously only known polynomial-time algorithm, running in $O(n^{18})$ time, to find a uv-trail. We reduce the complexity to $\tilde{O}(n^{2\omega})$ time, where the $\tilde{O}$ notation hides poly-logarithmic factors, leading to a largely improved $O(n^{4.75})$-time algorithm. This third result has appeared in *Proceedings of the 39th International Symposium on Theoretical Aspects of Computer Science* (STACS 2022).

Keywords: Perfect graph, Induced subgraph, Odd hole, Induced path, Non-shortest path, Dynamic data structure.

doi:10.6342/NTU202203481

簡介

圖論與演算法　一張圖 G 都藉由他的邊集 $E(G)$ 所描述其節點集 $V(G)$ 成員之間的相鄰關係。「圖論」(graph theory) 最早起源於尤拉 (Euler) 在 1736 年為解決七橋問題而寫的一篇文章 (可參考例如 [10])，在往後將近三百年的時間裡，學者們不斷探索各種圖類別的各樣性質。這學問在資工眾多領域中一直扮演極為關鍵的基礎角色，不管是理論上還是應用的離散組合問題，往往可以化約成圖論的問題來處理。其中演算法的研究則與圖論有著最密切的互動。因此圖論演算法在資訊科學的理論研究中歷久不衰。

圖運算　「透過一些運算由一張圖得到另一張圖」的概念淺顯易懂，卻蘊涵圖論最深奧的智慧。例如圖 H 是圖 G 的「子圖」(subgraph) 是指 G 可以透過拔點 (vertex deletion) 與拔邊 (edge deletion) 的運算得到 H, 也就是同時滿足 $V(H) \subseteq V(G)$ 與 $E(H) \subseteq E(G)$. 圖 H 是圖 G 的「次圖」(minor) 是指 G 可以透過拔點、拔邊以及縮邊 (edge contraction) 的運算得到 H. 學界公認最深奧的圖論成果之一，當屬 Robertson 與 Seymour 兩位圖論大師，歷經二十多年，用 22 篇 *Journal of Combinatorial Theory, Series B* (JCTB) 加上 1 篇 *Journal of Algorithms*, ，以「連載」的方式所證明完成的「次圖定理」(Graph Minor Theorem) [75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97]，解決 1970 年代 Wagner 所留下的懸案。這個被 2021 年數學界最高榮譽 Abel Prize 得主的 Lovász 推崇為 “a monumental project in graph theory” [69] 的偉大成就，正是以是否存在「次圖」的方式來刻劃非常重要的圖類別。

doi:10.6342/NTU202203481

導出子圖　H 是 G 的「導出子圖」(induced subgraph) 是指 G 能夠只透過拔點這個運算獲得 H, 所以導出子圖是一種特別的次圖。導出子圖這個概念與圖論中的諸多重要議題相關，其中最著名的例子是「完美圖」(perfect graph)，也就是該圖中所有導出子圖都符合著色數 (chromatic number) 等於最大團的頂點數 (clique number) 的圖類別。完美圖的概念在 1960 年代由 Berge 提出 [4, 5]，他當時就猜測完美圖與導出環 (induced cycle) 有著非常直接的關聯：這個被稱為 Berge conjecture 的猜測就是說「G 是完美圖若且為 G 與補圖 $\bar{G}$ 中都沒有奇數長度的『洞』(hole, 也就是長度至少為 5 的導出環」。這個猜測懸宕將近半個世紀才被 Chudnovsky, Robertson, Seymour 與 Thomas 四位圖論巨擘在數學界最頂尖的期刊 *Annals of Mathematics* 以一百七十頁的篇幅證出。這個猜測也被改稱為「強完美圖定理」(Strong Perfect Graph Theorem) [20]。這四位作者因此獲得 2009 年 Fulkerson Prize 的殊榮，其中最年輕的 Chudnovsky，是 Seymour 的得意門生，也因為這個傑出成就，在 2012 年獲得俗稱「天才獎」(Genius Grant) 的 MacArthur Award 的肯定。根據強完美圖定理，第一個在多項式時間內判別 n 個節點的圖是否為完美圖的演算法在 2003 年被提出，時間複雜度為 $O(n^{18})$ [39]，不久被改進到 $O(n^9)$ [17]。本篇論文的三個結果，皆是圍繞並改進圖論大師 Seymour 的演算法作品。Seymour 曾長期同時擔任圖論最頂尖的三個期刊 (1) *Journal of Combinatorial, Series B* (JCTB), (2) *Journal of Graph Theory*, 與 (3) *Combinatorica* 的主編. 他光是發表在 JCTB 的論文就已經有 114 篇。

在給定的圖中偵測符合某些透過各種圖運算所得獲得的圖類別，是所謂的圖偵測 (graph detection) 問題，既然子圖與次圖是圖論中最基礎卻深奧的概念，偵測子圖與次圖的自然是演算法領域中極其重要的研究主題。而其中導出子圖的偵測演算法，跟相對應的一般子圖的偵測演算法比起來難度往往高出非常多。例如剛出爐的 Dalirrooyfard 和 Vassilevska Williams 在 *IEEE FOCS* 2022 發表的最新論文 [42]，標題就叫做 “Induced cycles and paths are harder than you think”，她們證明要偵測圖中是否存在一個有 k 個節點的導出路徑或導出環的難度至少和偵測 $3k/4 - O(1)$ 個節點的團一樣難。接下來我們也進一步以三個最基本的圖類別：路徑、樹以及環來舉例，比較一般版與導出版的難度差別:

doi:10.6342/NTU202203481

路徑 比如要判斷 n 個節點的圖上三個節點 u, v, w 之間是否存在一條路徑 (path) 相接，是不難的演算法問題 [62]，但若額外要求連接三個節點的必須是導出路徑 (induced path)，難度立刻升高到 NP 完備的 three-in-a-path 問題 [54]。而要判斷兩個節點 u 與 v 之間是否存在一條非最短路徑也不難。一條第 k 短的路徑可以在近線性時間內被找到 [44]。但若要求此非最短路徑必須是導出路徑，目前文獻中最快的方法也是 Seymour 的作品，複雜度高達 $O(n^{18})$ [7]，而此篇論文的第三個結果就把時間大幅度地改進到 $O(\log^2 n)$ 次 n^2 階方陣所需的時間，例如引用剛剛發表在 SODA 2021 的矩陣乘法演算法 [2] 就能夠在 $O(n^{4.75})$ 時間內解決，改進的幅度超過 $\Theta(n^{13})$。此結果也已經發表於 STACS 2022 [16]。

樹 要判斷 n 個節點的圖上是否存在一棵樹連結一個任意的節點集合中的所有節點，我們知道這其實就是判定這些節點是否在同一個連通元件 (connected component) 中，可以很輕易的被找出來。但是若要求此樹必須是導出樹，則立刻變成 NP 完備的問題 [52]。

環 要判斷 n 個節點的圖上是否存在一個環通過兩個節點 u 與 v 能夠在 $O(n^2)$ 時間內解掉 [62]。但若要求此環必須為洞，那難度就立刻變為 NP 完備的 two-in-a-cycle 問題 [8]。而如果不要求通過任何節點，只判斷圖上是否存在一個環或洞，那兩者都可以在 $O(n^2)$ 時間內被找出。但若更進一步要求這些環或洞長度的奇偶性，那狀況就不同了。要判斷圖上是否存在一個偶環 (even cycle)，經過學者們不斷地競爭，早在 1997 (SIAM J. Discrete Math) 就已經有 $O(n^2)$ 時間的演算法 [105]，而且在同樣時間內還能找出最短的偶環。但是判斷圖上是否有偶洞 (even hole) 也就是導出偶環卻是直到 1997 年才在 IEEE FOCS 出現第一個多項式時間的演算法，複雜度高達 $O(n^{40})$ [29]。Chudnovsky, Kawarabayashi, Seymour 三人在 2005 年 (Journal of Graph Theory) [18] 將複雜度壓低到 $O(n^{31})$ 的同時，也明列找出最短的偶洞是個 open problem。同年 ACM Transactions on Algorithms 創刊號 Johnson [60] 在他那著名的 NP-complete column 中也把尋找最短偶洞的 open problem 列在當中，直到 Cheong and Lu 發表在 Journal of Graph Theory 2022 年的論文 [15] 才在 $O(n^{31})$ 時間內解決這個問題。至於偶洞偵測的問題，da Silva 等人 (JCTB 2013 [40]) 壓到 $O(n^{19})$，然後 Chang

doi:10.6342/NTU202203481

and Lu (JCTB 2015 [13]) 將複雜度壓到 $O(n^{11})$。Lai, Lu, and Thorup (STOC 2020 [65]) 則更壓低到目前最快的 $O(n^9)$。更有趣的是奇環 (odd cycle) 的偵測，也是數十年前就有快速演算法，但奇洞 (odd hole) 的偵測卻懸案到 2020 才被 Chudnovsky, Scott, Seymour, Spirkl 四人 (JACM 2020 [24]) 花 $O(n^9)$ 時間解掉。同年 Lai, Lu, and Thorup 的 STOC 2020 論文 [65] 也立即接著將這個問題的複雜度壓低到 $O(n^8)$。而本篇論文的第一個結果則更進一步的把複雜度壓低到 $O(n^7)$。根據強完美圖定理，這同時也把完美圖判別的時間複雜度壓低到 $O(n^7)$。找出最短的奇洞則是 2021 年才由 Chudnovsky, Scott, Seymour (ACM Transactions on Algorithms 2021 [23]) 花 $O(n^{14})$ 時間解決。本篇論文的第二個結果把這個時間壓低到 $O(n^{13})$。

doi:10.6342/NTU202203481

Contents

Acknowledgement **i**

摘要 **ii**

Abstract **iii**

中文簡介 **v**

Contents **ix**

1 Introduction **1**

2 Recognizing Perfect Graphs via Detecting Odd Holes **7**

2.1 Technical Overview 8

2.2 Proving Theorem 1 10

2.3 Proving Lemma 2.2 12

2.4 Proving Lemma 2.3 12

2.4.1 Proving Lemma 2.5 15

2.4.2 Proving Lemma 2.6 15

2.5 Proving Lemma 2.4 16

2.5.1 Proving Lemma 2.9 20

doi:10.6342/NTU202203481

2.5.2 Proving Lemma 2.10 20

3 Finding a Shortest Odd Hole **23**

3.1 Technical Overview 23

3.2 Proving Theorem 2 24

3.3 Proving Lemma 3.1 24

4 Finding a Non-shortest Induced Path **31**

4.1 Technical Overview 31

4.2 A simpler algorithm 33

4.3 Proof of Theorem 3 40

5 Conclusion **48**

References **50**

doi:10.6342/NTU202203481

Chapter 1

Introduction

Induced subgraphs are important Let G be an n-vertex undirected and unweighted graph. Let $V(G)$ consist of the vertices of G. For any graph H, let $G[H]$ be the subgraph of G induced by $V(H)$. A subgraph H of G is *induced* if $G[H] = H$. That is, an induced subgraph of G is a graph that can be obtained from G by deleting a set of vertices in tandem with its incident edges. To *detect* an (induced) graph H in G is to determine whether H is isomorphic to an (induced) subgraph of G. To *find* an (induced) graph H in G is to report an (induced) subgraph of G that is isomorphic to H, if there is one. Various classes of induced subgraphs are involved in the deepest results of graph theory and graph algorithms. One of the most prominent examples concerns the *perfection* of G that the chromatic number of each induced subgraph H of G equals the clique number of H. A graph is *odd* (respectively, *even*) if it has an odd (respectively, even) number of edges. A *hole* of G is an induced cycle of G having at least four edges. The seminal Strong Perfect Graph Theorem of Chudnovsky, Robertson, Seymour, and Thomas [20, 26], conjectured by Berge in 1960 [4, 5, 6], confirms that the perfection of a graph G can be determined by detecting odd holes in G and its complement.

Based on the theorem, the first known polynomial-time algorithms for recognizing perfect graphs

doi:10.6342/NTU202203481

take $O(n^{18})$ [39] and $O(n^9)$ [17] time. The $O(n^9)$-time version can be implemented to run in $O(n^{6+\omega})$ time [65, §6.2] via efficient algorithms for the *three-in-a-tree* problem [25] that detects induced subtrees of G spanning three prespecified vertices, where $\omega < 2.373$ [2, 38, 66, 103] is the exponent of square-matrix multiplication.

Detecting induced subgraphs is challenging Detecting induced subgraphs, even the most basic ones like paths, trees, and cycles, is usually more challenging than detecting their counterparts that need not be induced [42]. For instance, detecting a path spanning three prespecified vertices is tractable (via, e. g., [62, 87]), but the *three-in-a-path* problem that detects an induced path spanning three prespecified vertices is NP-hard (see, e. g., [54, 65]).

Detecting trees spanning a given set of vertices is easy via the connected components, but detecting induced trees spanning a set of prespecified vertices is NP-hard [52]. The three-in-a-tree problem is shown to be solvable first in $O(n^4)$ time [25] and then in $\tilde{O}(n^2)$ time [65] via involved structural theorems and dynamic data structures. The tractability of the corresponding k-in-a-tree problem for any fixed $k \geq 4$ is still unknown, although the problem can be solved in $O(n^4)$ time on a graph of girth at least k [67].

Detecting odd holes Cycle detection has a similar situation. Detecting cycles of length three, which have to be induced, is the classical triangle detection problem that can be solved efficiently by matrix multiplications (see, e. g., [104]). It is tractable to detect cycles of length at least four spanning two prespecified vertices (via, e. g., [62, 87]), but the *two-in-a-cycle* problem that detects holes spanning two prespecified vertices is NP-hard (and so are the corresponding one-in-an-even-cycle and one-in-an-odd-cycle problems) [8, 9]. See, e. g., [73, §3.1] for graph classes on which the two-in-a-cycle problem is tractable.

Detecting cycles without the requirement of spanning prespecified vertices is straightforward. Even and odd cycles are also long known to be efficiently detectable (see, e. g., [3, 41, 105]). It takes

doi:10.6342/NTU202203481

an $O(n^2)$-time depth-first search to detect odd cycles even if the graph is directed (see, e. g., [11, Table 1]). While detecting holes (i. e., recognizing chordal graphs) is solvable in $O(n^2)$ time [99, 100, 101], detecting odd (respectively, even) holes is more difficult. There are early $O(n^3)$-time algorithms for detecting odd and even holes in planar graphs [57, 72], but the tractability of detecting odd holes was open for decades (see, e. g., [27, 30, 33]) until the recent major breakthrough of Chudnovsky, Scott, Seymour, and Spirkl [24]. Their $O(n^9)$-time algorithm is later implemented to run in $O(n^8)$ time [65], immediately implying the best formerly known algorithm for recognizing perfect graphs based on the Strong Perfect Graph Theorem. Finding an odd hole based on Lai et al.'s $O(n^8)$-time algorithm for detecting odd holes takes $O(n^9)$ time. We improve the time of detecting and finding odd holes and recognizing perfect graphs to $O(n^7)$.

Theorem 1. *For an n-vertex m-edge graph G,*

(1) it takes $O(mn^5)$ time to either obtain an odd hole of G or ensure that G is odd-hole-free and, hence,

(2) it takes $O(n^7)$ time to determine whether G is perfect.

Finding a shortest odd hole A shortest cycle of G can be found in $\tilde{O}(n^\omega)$ time (even if G is directed) [58]. The time becomes $O(n)$ when G is planar [12]. A shortest odd cycle of G can be found in $O(n^3)$ time even if G is directed (see, e. g., [11, §1]). However, the previously only known polynomial-time algorithm to find a shortest odd hole of G takes $O(n^{14})$ time [23]. We further reduce the required time to $O(n^{13})$.

Theorem 2. *For an n-vertex m-edge graph, it takes $O(m^3n^7)$ time to either obtain a shortest odd hole of G or ensure that G is odd-hole-free.*

Detecting even holes As for detecting even holes, the first polynomial-time algorithm, running in about $O(n^{40})$ time, appeared in 1997 [29, 31, 32]. It takes a line of intensive efforts to bring down the complexity to $O(n^{31})$ [18], $O(n^{19})$ [40], $O(n^{11})$ [13], and finally $O(n^9)$ [65]. A shortest

doi:10.6342/NTU202203481

even cycle of G is long known to be computable in $O(n^2)$ time [105]. Very recently, a shortest even cycle of a directed G is shown to be obtainable in $\tilde{O}(n^{4+\omega})$ time with high probability via an algebraic approach [11]. On the other hand, the tractability of finding a shortest even hole, open for 16 years [18, 60], is resolved by a newly announced $O(n^{31})$-time algorithm [15]. See [22] (respectively, [37] for detecting an odd (respectively, even) hole with a prespecified length lower bound. See [1, 19] for the first polynomial-time algorithm for finding an independent set of maximum weight in a graph having no hole of length at least five. See [43] for upper and lower bounds on the complexity of detecting an $O(1)$-vertex induced subgraph.

Detecting non-shortest induced paths The *two-in-a-path* problem that detects induced paths spanning two prespecified vertices is equivalent to determining whether the two vertices are connected. Nonetheless, the corresponding two-in-an-odd-path and two-in-an-even-path problems are NP-hard [8, 9], whose state-of-the-art algorithms on a planar graph take $O(n^7)$ time [61]. See [45, 47, 71] for how an induced even uv-path of G affects the perfection of G. See [64] for a conjecture by Erdős on how an induced uv-path of G affects the connectivity between u and v in G. Finding a longest uv-path in G that has to (respectively, need not) be induced is NP-hard [49, GT23] (respectively, [49, ND29]). See [51, 59] for longest or long induced paths in special graphs. The presence of long induced paths in G affects the tractability of coloring G [50]. See also [1] for the first polynomial-time algorithm for finding a minimum feedback vertex set of a graph having no induced path of length at least five. Detecting a non-shortest uv-path in G is easy. A k-th shortest uv-path in G can also be found in near linear time [44]. See [55] for listing induced paths and holes. See [14, §4] for the parameterized complexity of detecting an induced path with a prespecified length. Detecting an induced uv-path in a directed graph G is NP-complete (even if G is planar) [46] and $W[1]$-complete [54]. However, the tractability of detecting a non-shortest induced uv-path in an undirected graph G was unknown until the recent result of Berger, Seymour, and Spirkl [7].

doi:10.6342/NTU202203481

Let $\|G\|$ denote the number of edges in G. A path with end-vertices u and v is a *uv-path*. A uv-path P of G is *shortest* if G admits no uv-path Q with $\|Q\| < \|P\|$, so each shortest uv-path of G is induced. We call an induced uv-path of G that is not a shortest uv-path of G a *uv-trail* of G. A graph admitting no uv-trail is *uv-trailless*. Berger, Seymour, and Spirkl [7] gave the formerly only known polynomial-time algorithm, running in $O(n^{18})$ time, to either output a uv-trail of G or ensure that G is uv-trailless. Their result leads to an $O(n^{21})$-time algorithm [36] to determine whether all holes of G have the same length. We improve the time of finding a uv-trail to $O(n^{4.75})$ as summarized in the following theorem, where the $\tilde{O}$ notation hides poly-logarithmic factors.

Theorem 3. *For any two vertices u and v of an n-vertex graph G, it takes $\tilde{O}(n^{2\omega})$ time to either obtain a uv-trail of G or ensure that G is uv-trailless.*

Theorem 3 immediately reduces the $O(n^{21})$ time of recognizing a graph with all holes the same length to $O(n^{7.75})$. Theorem 3 has appeared in STACS 2022 [16].

General approach Our three algorithms to find induced subgraphs of G are based on the following "guess-and-verify" approach, which has been extensively applied in the literature of algorithms for induced subgraphs (see, e. g., [7, 17, 23, 24, 65]): For each choice of guessed ℓ vertices, run an $O(f(n))$-time subroutine on the vertex ℓ-tuple. If a target induced subgraph H of G is found, then report H. Otherwise, if all $O(n^\ell)$ vertex ℓ-tuples are tested and nothing is reported, then report that G does not contain any such induced subgraphs. This is an $O(n^\ell \cdot f(n))$-time algorithm to find an induced subgraph of G.

1. Our proof of Theorem 1 includes an $O(n^7)$-time bottleneck task that runs an $O(n^2)$-time subroutine for each choice of 5 guessed vertices.
2. Our proof of Theorem 2 includes an $O(n^{13})$-time bottleneck task that runs an $O(n^2)$-time subroutine for each choice of 11 guessed vertices.
3. Our proof of Theorem 3 includes an $O(n^4 \log n)$-time task that runs an $O(n^2 \log n)$-time sub-

doi:10.6342/NTU202203481

routine for each choice of 2 guessed vertices.

Preliminaries and roadmap For integers i and k, let $[i, k]$ consist of the integers j with $i \leq j \leq k$ and let $[k] = [1, k]$. Let $|S|$ denote the cardinality of a set S. Let $R \setminus S$ for sets R and S consist of the elements of R that are not in S. Let $E(G)$ for a graph G consist of the edges of G and thus $|E(G)| = \|G\|$. A *k-graph* (e. g., 2-path or 5-hole) is a graph having k edges. A *triangle* is a 3-cycle. The *length* of a path or a cycle is its number of edges. Let $H \subseteq G$ for a graph H denote $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Let $G - V$ for a set V of vertices denote $G[V(G) \setminus V]$. Let $G - v$ for a vertex v be $G - \{v\}$. Let $G \setminus E$ for a set E of edges denote the graph obtained from G by deleting its edges in E. For any $u \in V(G)$, let $N_G(u)$ consist of the vertices v with $uv \in E(G)$ and $N_G[u] = \{u\} \cup N_G(u)$. A *leaf* of a graph G is a degree-1 vertex of G. Let *int*(P) consist of the interior vertices of a path P. A *UV-path* for vertex sets U and V is a uv-path with $u \in U$ and $v \in V$. Let $T[u, v]$ with $\{u, v\} \subseteq V(T)$ for a tree T denote the simple uv-path of T. If vertices u and v of G are connected in G, then let $d_G(u, v)$ denote the length of a shortest uv-path of G. Otherwise, let $d_G(u, v) = \infty$. For any graph H, let $N_G(H)$ consist of the vertices $v \notin V(H)$ with $uv \in E(G)$ for some $u \in V(H)$ and $N_G[H] = V(H) \cup N_G(H)$. For any graphs D and H, let $N_G(u, D) = N_G(u) \cap V(D)$ and $N_G(H, D) = N_G(H) \cap V(D)$. Graphs H and D are *adjacent* (respectively, *anticomplete*) in G if $N_G(H, D) \neq \varnothing$ (respectively, $N_G[H] \cap V(D) = \varnothing$).

It is convenient to assume that the n-vertex m-edge graph G of Theorems 1, 2, and 3 are connected for the rest of the thesis , which is organized as follows. Chapter 2 proves Theorem 1. Chapter 3 proves Theorem 2. Chapter 4 proves Theorem 3. Chapter 5 concludes the thesis.

doi:10.6342/NTU202203481

Chapter 2

Recognizing Perfect Graphs via Detecting Odd Holes

Perfect graphs are important objects for graph theory. See [102] for a survey. Although the graph coloring problem, maximum clique problem, and maximum independent set problem are all NP-complete in general graphs, they are all tractable in perfect graphs [53] via the ellipsoid algorithm [63]. In 1960, Berge first introduced the concept of perfect graphs. He also proposed two conjectures:

(1) A graph G is perfect if and only if its complement $\bar{G}$ is perfect.

(2) A graph G is perfect if and only if both G and $\bar{G}$ are odd-hole-free.

Since (2) implies (1), (1) is called the (weak) perfect graph conjecture and (1) is called the strong perfect graph conjecture. (1) was proved by Lovász in 1972 [68] and thus called the perfect graph theorem. (2) had been extensively studied over 4 decades and remained open until proved by Chudnovsky et al. in 2006. Thus, (2) is called the strong perfect graph theorem. We give a brief overview of the history of proving the strong perfect graph theorem. Since a perfect graph is Berge (see, e. g., [102]), it remains to prove the converse to show (2). Most of the work on (2) falls into

doi:10.6342/NTU202203481

3 groups:

1. Proving that (2) holds for graphs with some induced subgraphs excluded (see, e. g., [28]),
2. Investigating the structure of minimum imperfect graphs, which is graphs that is not perfect with minimum vertices (see, e. g., [21]), and
3. Showing that every Berge graph either belongs to some ”basic” classes of perfect graphs or admits some feature that an imperfect graph can not admit.

Chudnovsky et al.'s proof adopts the third idea. Their proof was mainly inspired by a conjecture [35] of Conforti, Cornuéjols, and Vušković in 2004, which also adopts the third idea. Specifically, Chudnovsky et al.'s proof idea is showing that every Berge graph either belongs to 5 basic classes of perfect graphs or admits one of 4 kinds of decomposition into simpler subgraphs. These kinds of decomposition are designed so that a minimum imperfect graph can not admit any kinds of such decomposition. Similar ideas also appeared in other work on perfect graphs and inspired Chudnovsky et al. (see, e. g., [34, 74]). The idea of "some graphs either fall into one of a few basic classes or admit a decomposition" is also used in the proof of other theorems in graph theory (see, e. g., [28, 70, 98]). Based on the strong perfect graph threorem, the perfection of G can be determined by detecting odd holes in G and its complement. We first give a technical overview of the proof of Theorem 1 in §2.1 and prove it in the rest of the chapter.

2.1 Technical Overview

The first known polynomial-time algorithm of Chudnovsky, Scott, Seymour, and Spirkl [24] for detecting odd holes consists of four subroutines:

(1) Detecting "jewels" in $O(n^6)$ time [17, 3.1].

(2) Detecting "pyramids" in $O(n^9)$ time [17, 2.2].

(3) Detecting "heavy-cleanable" shortest odd holes in a graph having no jewel and pyramid in

doi:10.6342/NTU202203481

$O(n^8)$ time [24, Theorem 2.4].

(4) Detecting odd holes in a graph having no jewel, pyramid, and heavy-cleanable shortest odd hole in $O(n^9)$ time [24, Theorem 4.7].

Lai, Lu, and Thorup [65] improve the complexity to $O(n^8)$ by reducing the time of (2), (3), and (4) to $\tilde{O}(n^5)$ [65, Theorem 1.3], $O(n^5)$ [65, Lemma 6.8(2)], and $O(n^8)$ [65, Proof of Theorem 1.4], respectively. Finding odd holes based on Chudnovsky et al.'s $O(n^9)$-time (respectively, Lai et al.'s $O(n^8)$-time) algorithm for detecting odd holes takes $O(n^{10})$ (respectively, $O(n^9)$) time. We further improve the time of detecting and finding odd holes to $O(n^7)$ by the following arrangement.

- Extending the concept of a graph containing jewels (respectively, heavy-cleanable shortest holes and pyramids) to that of a shallow (respectively, medium and deep) graph (defined in §2.2).
- Generalizing
  - (1) to an $O(n^7)$-time subroutine for finding a shortest odd hole in a shallow graph (Lemma 2.2),
  - (2) to an $\tilde{O}(n^6)$-time subroutine for finding an odd hole in a deep graph (Lemma 2.1), and
  - (3) to an $O(n^5)$-time subroutine for finding a shortest odd hole in a non-shallow, medium, and non-deep graph (Lemma 2.3).
- Specializing
  - (4) to an $O(n^7)$-time subroutine for finding a shortest odd hole in a non-shallow, non-medium, and non-deep graph (Lemma 2.4).

Chudnovsky et al.'s $O(n^9)$-time subroutine for (4) has six procedures. The i-th procedures with $i \in \{1, 2\}$ (respectively, $i \in \{3, \ldots, 6\}$) enumerate all $O(n^6)$ six-tuples $x = (x_0, \ldots, x_5)$ (respectively, $O(n^7)$ seven-tuples $x = (x_0, \ldots, x_6)$) of vertices and spend $O(n^3)$ (respectively, $O(n^2)$) time for each x to examine whether there is an odd hole of the i-th type that contains all vertices of x other than x_0. Lai et al.'s $O(n^8)$-time subroutine for (4) achieves the improvement by

(a) reducing the number of enumerated vertices to five and keeping the examination time in $O(n^3)$ for the i-th procedures with $i \in \{1, 3, 5\}$ and

doi:10.6342/NTU202203481

(b) keeping the number of enumerated vertices in six and reducing the examination time to $O(n^2)$ for the i-th procedures with $i \in \{2, 4, 6\}$.

Our specialized $O(n^7)$-time subroutine for (4) is based on a new observation that at most five of the vertices in x suffice for each of the six procedures to pin down an odd hole. Skipping a vertex (i.e., x_1 or x_2 in the proof of Lemma 2.4) to reduce the number of rounds from $O(n^6)$ to $O(n^5)$ complicates the task of examining the existence of an odd hole containing the remaining five vertices other than x_0. We manage to complete the task within the same $O(n^2)$ time bound via some data structures.

2.2 Proving Theorem 1

The rest of the chapter assumes without loss of generality that G contains no 5- or 7-hole, which can be listed in $O(mn^5)$ time. A $D \subseteq V(G)$ with $|D| \leq 5$ is a *spade* for a shortest odd hole C of G if

- $C[D]$ is a uv-path,
- $G[D]$ contains an induced uv-path with length $\|C[D]\| + 1$ or $\|C[D]\| - 1$, and
- $C - B$ with $B = N_G[D \setminus \{u, v\}] \setminus \{u, v\}$ is a shortest uv-path of $G - B$.

For instance, if $C[N_G(x)]$ is a 3-path for an $x \in V(G)$, then $N_G(x, C) \cup \{x\}$ is a spade for C. A hole C of G is *shallow* if C is a shortest odd hole of G that admits a spade. We comment that a jewelled [23] shortest odd hole of G need not be a shallow hole of G but implies a shallow hole of G. Let $M_G(C)$ consist of the (major [18]) vertices x of G such that $N_G(x, C)$ is not contained by any 2-path of C. A hole C of G is *medium* if C is a shortest odd hole of G and $M_G(C) \subseteq N_G(e)$ holds for an $e \in E(C)$. Thus, 5-holes are medium. A medium hole is a heavy-cleanable shortest odd hole in [23]. A triple $T = (T_1, T_2, T_3)$ of ab_i-paths T_i for $i \in [3]$ with $\|T_1\| < \|T_2\| \leq \|T_3\|$ is a *tripod* of G if $\|T_1\|$ is minimized over all triples T satisfying the following *Conditions Z*:

doi:10.6342/NTU202203481

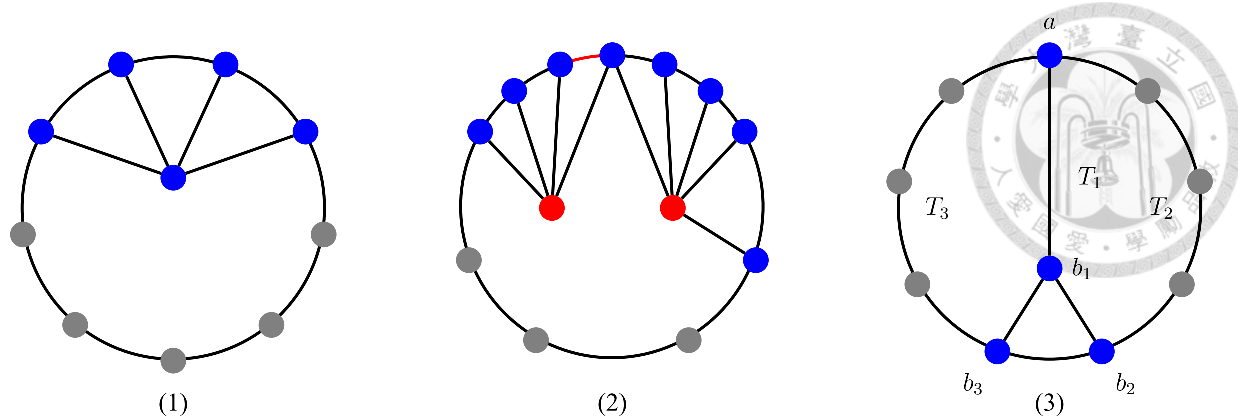


Figure 2.1: (1) The black circle denotes a shallow hole C of the graph. The blue vertices denote a spade D for C. (2) The black circle denotes a medium hole C of the graph. The red vertices denote the vertices in $M_G(C)$. The red edge is adjacent to each vertex in $M_G(C)$. (3) The black circle denotes a deep hole C of the graph. $T = (T_1, T_2, T_3)$ is a tripod of the graph with $C = C(T)$.

Z1: $B(T) = \{b_1, b_2, b_3\}$ induces a triangle of G.

Z2: $U(T) = T_1 \cup T_2 \cup T_3$ is an induced tree of $G \setminus E(G[B(T)])$ with the leaf set $B(T)$.

Z3: $a(T) = a$ is the only degree-3 vertex of $U(T)$.

Z4: $C(T) = G[T_2 \cup T_3]$ is a shortest odd hole of G.

A hole of G is *deep* if it is $C(T)$ for a tripod T of G. Such a $G[U(T)]$ is called an optimal great pyramid of G with apex $a(T)$ and base $B(T)$ in [23]. A graph is *shallow* (respectively, *medium* and *deep*) if it contains a shallow (respectively, medium and deep) hole. The *depth* δ_G of a deep graph G is $\|T_1\|$ for a tripod T of G. See Figure 2.1 for an illustration.

Lemma 2.1 (Lai, Lu, and Thorup [65, Theorem 1.3])**.** *It takes $O(mn^4 \log^2 n)$ time to obtain a $C \subseteq G$ such that (1) C is an odd hole of G or (2) G is non-deep.*

Lemma 2.2. *It takes $O(mn^5)$ time to obtain a $C \subseteq G$ such that (1) C is a shortest odd hole of G or (2) G is non-shallow.*

Lemma 2.3. *It takes $O(mn^3)$ time to obtain a $C \subseteq G$ such that (1) C is a shortest odd hole of G or (2) G is shallow, deep, or non-medium.*

doi:10.6342/NTU202203481

Lemma 2.4. *It takes $O(mn^5)$ time to obtain a $C \subseteq G$ such that (1) C is a shortest odd hole of G or (2) G is shallow, medium, deep, or odd-hole-free.*

Lemma 2.2 corresponds to the algorithm for jewelled shortest odd holes in [23, 2.1]. Lemma 2.3 improves on the $O(m^2n^4)$-time algorithm of [23, 6.2]. Lemma 2.4 improves on the $O(m^2n^5)$-time algorithm of [23, 6.3] and the $O(m^2n^4)$-time algorithm in [65, Proof of Theorem 1.4]. We reduce Theorem 1 to Lemmas 2.2, 2.3, and 2.4 via Lemma 2.1. Lemmas 2.2, 2.3, and 2.4 are proved in §2.3, §2.4, and §2.5.

Proof of Theorem 1. It suffices to prove (1). It takes $O(m)$ time to determine if one of the four C is an odd hole of G. If there is one, then (1) holds. Otherwise, G is non-deep by Lemma 2.1, non-shallow by Lemma 2.2, and non-medium by Lemma 2.3, implying that G is odd-hole-free by Lemma 2.4. □

2.3 Proving Lemma 2.2

Proof of Lemma 2.2. It takes $O(m)$ time to determine for each $D \subseteq V(G)$ with $|D| \leq 5$ whether G contains odd holes for which D is a spade. If G contains such odd holes, then let C_D be a shortest of them. Otherwise, let $C_D = \varnothing$. If all C_D are empty, then let the $O(mn^5)$-time obtainable C be empty. Otherwise, let C be a non-empty C_D with minimum $\|C_D\|$. If G contains a shallow hole C^*, then $0 < \|C\| \leq \|C_D\| \leq \|C^*\|$ holds for a spade D for C^*, implying that C is a shortest odd hole of G. □

2.4 Proving Lemma 2.3

A *clean* hole of G is a medium hole C of G with $M_G(C) = \varnothing$.

Lemma 2.5. *If G is a non-deep (respectively, non-shallow) graph, then so is an induced subgraph*

doi:10.6342/NTU202203481

of G that contains a shortest odd hole of G.

Lemma 2.6 (Chudnovsky, Scott, and Seymour [23, Proof of Lemma 6.1])**.** *If u and v are vertices of a clean hole C of a non-shallow and non-deep graph H, then the graph obtained from C by replacing the shortest uv-path of C with a shortest uv-path of H remains a clean hole of H.*

We first reduce Lemma 2.3 to Lemma 2.5 via Lemma 2.6 and then prove Lemma 2.5 in §2.4.1. We also include a proof of Lemma 2.6 in §2.4.2 to ensure that it is implicit in [23, Proof of Lemma 6.1]. Lemma 2.6 is stronger than [17, Theorem 4.1(2)] in that G is allowed to contain jewels or pyramids. As a matter of fact, the original proof of [17, Theorem 4.1(2)] already works for Lemma 2.6: Their careful case analysis shows that if the resulting subgraph is not a clean hole of G, then G contains a jewel or pyramid. It is not difficult to further infer that each such jewel (respectively, pyramid) in G contains a shallow (respectively, deep) hole of G.

Proof of Lemma 2.3. (Inspired by [65, Proof of Lemma 6.8(2)].) For each $e \in E(G)$ and $u \in V(G)$, spend $O(m)$ time to obtain a shortest-path tree of $G - N_G(e) \setminus \{u\}$ rooted at u, from which spend $O(n)$ time for each $v \in V(G)$ to obtain a shortest uv-path $P_e(v, u)$, if any, of $G_e(u, v) = G - (N_G(e) \setminus \{u, v\})$. Let $P_e(u, v) = P_e(v, u)$ for each $\{u, v\} \subseteq V(G)$ without loss of generality. Thus, it takes overall $O(mn^3)$ time to obtain for all edges e and distinct vertices u and v of G with defined $P_e(u, v)$ (i) $p_e(u, v) = \|P_e(u, v)\|$ and (ii) the neighbor $\tau_e(u, v)$ of u in $P_e(u, v)$. Let $p_e(u, v) = \infty$ for undefined $P_e(u, v)$. Spend $O(mn^3)$ time to determine if the next equation holds for any edge e and distinct vertices $b, c,$ and d of G:

$$\begin{aligned}
p_e(c, d) &= 3 \\
p_e(c, \tau_e(d, b)) &> 3 \\
p_e(d, \tau_e(c, b)) &> 3 \\
p_e(c, b) = p_e(d, b) &= p_e(c, \tau_e(b, d)) - 1 = p_e(d, \tau_e(b, c)) - 1.
\end{aligned} \tag{2.1}$$

doi:10.6342/NTU202203481

If Equation (2.1) holds for some (e, b, c, d), then let $C = P_e(b, c) \cup P_e(b, d) \cup P_e(c, d)$ for such an (e, b, c, d) that minimizes $p_e(b, c) + p_e(b, d) + p_e(c, d)$. Otherwise, let $C = \varnothing$.

We show that if a non-shallow and non-deep graph G contains a medium hole C^*, then the above reported graph C is a shortest odd hole of G. Let e be an edge of C^* with $M_G(C^*) \subseteq N_G(e)$. C^* is a clean hole of the non-shallow and non-deep graph $H = G_e(c, d)$ with $\{c, d\} = N_{C^*}(e)$ by Lemma 2.5. For each $\{u, v\} \subseteq V(C^*)$ such that $\{c, d\}$ is disjoint from the interior of the shortest uv-path of C^*, $P_e(u, v)$ is a shortest uv-path of H. Therefore, for the vertex $b \in V(C^*)$ with $d_{C^*}(b, c) = d_{C^*}(b, d)$, Lemma 2.6 implies that $P_e(b, c) \cup P_e(b, d) \cup P_e(c, d)$ is a clean hole of H and hence a shortest odd hole of G. One can verify from $\|C^*\| \geq 9$ that Equation (2.1) holds for this (e, b, c, d). Thus, $C \neq \varnothing$. It remains to show that Equation (2.1) for any choice of (e, b, c, d) implies that $P_e(b, c) \cup P_e(b, d) \cup P_e(c, d)$ is an odd hole of G with length $p_e(b, c) + p_e(b, d) + p_e(c, d)$: Both $P_e(b, c)$ and $P_e(b, d)$ are induced paths. By

$$p_e(c, b) = p_e(d, b) = p_e(c, \tau_e(b, d)) - 1 = p_e(d, \tau_e(b, c)) - 1,$$

paths $P_e(b, c) - b$ and $P_e(b, d) - b$ are anticomplete in G. We know that $\text{int}(P_e(c, d))$ is anticomplete to $(P_e(c, b) - c) \cup (P_e(d, b) - d)$, since otherwise Equation (2.1) is violated by at least one of the following conditions:

$$\begin{aligned}
&p_e(c, \tau_e(d, b)) \leq 3 \\
&p_e(d, \tau_e(c, b)) \leq 3 \\
&p_e(c, \tau_e(b, d)) \leq p_e(c, b) \\
&p_e(d, \tau_e(b, c)) \leq p_e(d, b).
\end{aligned}$$

□

doi:10.6342/NTU202203481

2.4.1 Proving Lemma 2.5

Proof of Lemma 2.5. Let H be an induced subgraph of G that contains a shortest odd hole of G. Consider first the case that H is deep, implying that $C(T)$ is a shortest odd hole of G for a tripod T of G. All Conditions Z of T also hold in G, since H is an induced subgraph of G. If T is also a tripod of G, then G is deep. Otherwise, we have $\delta_G < \|T_1\|$, also implying that G is deep (so that δ_G is defined).

Consider now the case that H contains a shallow hole C for which D is a spade, implying that C is a shortest odd hole of G. Let $C[D]$ be a uv-path. $H[D]$ contains an induced uv-path R with length $\|C[D]\| + 1$ or $\|C[D]\| - 1$. Hence, $G[D] = H[D]$ contains an induced uv-path $Q \in \{C[D], R\}$ such that the union C^* of $Q = C^*[D]$ and a shortest uv-path P of $G - N_G[D \setminus \{u, v\}] \setminus \{u, v\}$ is an odd hole of G. Since $G[D]$ contains an induced uv-path, i.e., $C[D]$ or R with length $\|C^*[D]\| + 1$ or $\|C^*[D]\| - 1$, D is a spade for C^* in G. Since $H - N_H[D \setminus \{u, v\}] \setminus \{u, v\}$ is an induced subgraph of $G - N_G[D \setminus \{u, v\}] \setminus \{u, v\}$, we have $\|P\| \leq \|C\| - \|C[D]\|$. By $\|C^*[D]\| \leq \|C[D]\| + 1$, C^* is a shortest odd hole of G. Thus, C is a shallow hole of G. $\square$

2.4.2 Proving Lemma 2.6

Let $M^*_G(C) = \{x \in M_G(C) : |N_G(x, C)| \geq 4\}$, whose members are called the big major vertices for C in [23]. A path P of a graph G is C-*clean* for a shortest odd hole C of G if $M^*_G(C) \cap V(P) = \varnothing$.

Lemma 2.7 (Chudnovsky, Scott, and Seymour [23, 4.1])**.** *Let $\{u, v\} \subseteq V(C)$ for a shortest odd hole C of a non-shallow graph H. If P is a C-clean uv-path of H with $\|P\| < d_C(u, v)$, then H is deep with $\delta_H < \|P\|$.*

Lemma 2.8 (Chudnovsky, Scott, and Seymour [23, 4.2 and 4.3])**.** *Let $\{u, v\} \subseteq V(C)$ for a shortest odd hole C of a non-shallow non-deep graph H. If P is a C-clean uv-path of H with $\|P\| =$*

doi:10.6342/NTU202203481

$d_C(u,v)$, then the graph C^* obtained from C by replacing the shortest uv-path of C with P is a shortest odd hole of H with $M_H(C^*) = M_H(C)$.

Proof of Lemma 2.6. (Included to ensure that the lemma is implicit in [23].) By $M_H(C) = \varnothing$, P is C-clean. Since H contains the shortest uv-path Q of C, $\|P\| \le \|Q\|$. Lemma 2.7 implies $\|P\| = \|Q\|$, since H is non-deep. By Lemma 2.8, the graph C^* obtained from C by replacing Q with P is a shortest odd hole of H with $M_H(C^*) = \varnothing$. □

2.5 Proving Lemma 2.4

Lemma 2.9. *A shortest odd hole C of G with $M^*_G(C) \ne M_G(C)$ implies that G is deep with $\delta_G = 1$.*

Lemma 2.10. *If C is a non-shallow shortest odd hole of G, then each $x \in M^*_G(C)$ admits an $e \in E(C)$ with $M^*_G(C) \subseteq N_G(e) \cup N_G(x)$.*

Lemma 2.10 is stronger than [23, Theorem 5.3] in that G can be shallow. We first reduce Lemma 2.4 to Lemmas 2.9 and 2.10 via Lemmas 2.5 and 2.6. We then prove Lemmas 2.9 and 2.10 in §2.5.1 and §2.5.2.

Proof of Lemma 2.4. We first show an $O(n^2)$-time two-case subroutine that obtains a graph for each

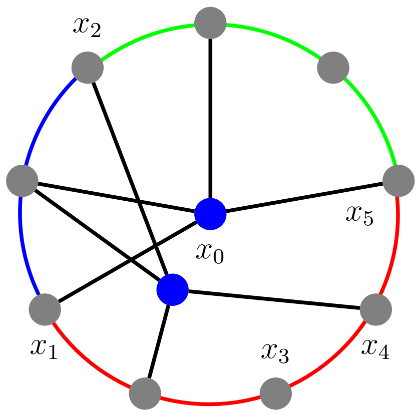
$$\{x_0, x_j, x_3, x_4, x_5\} \subseteq V(G)$$

with $j \in [2]$ and $x_4x_5 \in E(G)$ and each $k \in [3,5]$. If all $O(mn^3)$ of them are empty, then let the $O(mn^5)$-time obtainable graph C be empty. Otherwise, let C be a shortest of the nonempty ones. We then prove that C is a shortest odd hole of a non-shallow, non-medium, and non-deep G.

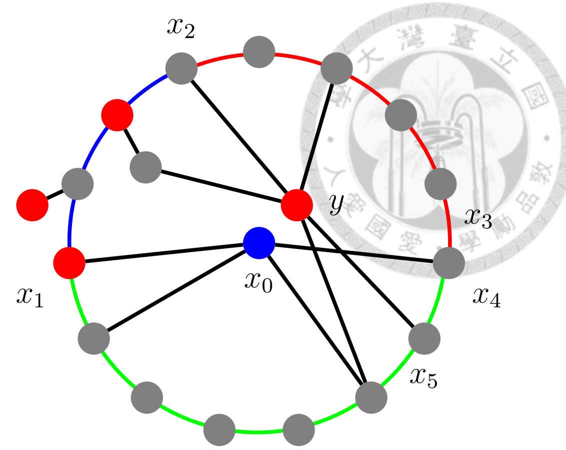
Case 1: $j = 1$. Let P be a shortest x_1x_k-path of the graph

$$H = G - (N_G[\{x_0, x_4, x_5\}] \setminus \{x_1, x_3, x_4, x_5\}),$$

doi:10.6342/NTU202203481



(1)



(2)

Figure 2.2: (1) An example for the proof of Lemma 2.4 with $j = 1$ and $k = 5$. The red arc is the shortest x_1x_k-path P of H. The blue arc is $P_1(x_2)$ and the green arc is $P_k(x_2)$. $C_1(x_2) = P \cup P_1(x_2) \cup P_k(x_2)$ is a shortest odd hole of (1). (2) An example for the proof of Lemma 2.4 with $j = 2$ and $k = 4$. The red arc is the shortest x_2x_k-path P of H. The red vertices denote the vertices in X_1. Although $y \notin V(H_1)$ and $y \notin I_1$, we have $y \in X_1$. Although $y \in V(G_1)$, we have $y \notin V(G_0(x_1))$. The blue arc is $P_2(x_1)$ and the green arc is $P_k(x_1)$. $C_2(x_1) = P \cup P_2(x_1) \cup P_k(x_1)$ is a shortest odd hole of (2).

as illustrated by Figure 2.2(1). Let I consist of the interior vertices of all shortest x_1x_k-paths of H. Let

$$G_0 = G - ((N_G(x_1) \cap N_G(x_k)) \cup (N_G[I] \setminus \{x_1, x_k\})).$$

Spend overall $O(n^2)$ time to obtain for each $i \in \{1, k\}$ and $v \in V(G_0)$ an arbitrary, if any, shortest x_iv-path $P_i(v)$ of G_0 and $R_i(v) = N_{G_0}[P_i(v) - v]$. For each $v \in V(G)$, it takes $O(n)$ time to determine if

$$C_1(v) = P \cup P_1(v) \cup P_k(v)$$

is an odd hole of G via $\|P\| + \|P_1(v)\| + \|P_k(v)\| \equiv 1 \pmod 2$ and $R_1(v) \cap V(P_k(v)) = \{v\}$. If none of the $O(n)$ graphs $C_1(v)$ is an odd hole of G, then report the empty graph. Otherwise, report a shortest one of the graphs $C_1(v)$ that are odd holes.

doi:10.6342/NTU202203481

Case 2: $j = 2$. Let P be a shortest x_2x_k-path of the graph

$$H = G - (N_G[\{x_0, x_4, x_5\}] \setminus \{x_3, x_4, x_5\}),$$

as illustrated by Figure 2.2(2). Let I consist of the interior vertices of all shortest x_2x_k-paths of H. With

$$H_1 = G - (N_G[\{x_0, x_4, x_5\} \cup I] \setminus \{x_2\}),$$

let I_1 consist of the vertices v with $d_{H_1}(v, x_2) \leq \|P\| - 3$. With $X_1 = N_G(I_1)$ and

$$G_1 = G - ((N_G(X_1) \cap N_G(x_k)) \cup (N_G(I_1 \cup I) \setminus (X_1 \cup \{x_2, x_3, x_4, x_5\}))),$$

let each $P_i(v)$ with $i \in \{2, k\}$ and $v \in V(G)$ be a shortest x_iv-path of the graph

$$G_0(v) = G_1 - (X_1 \setminus \{v\}).$$

It takes overall $O(n^2)$ time to determine whether $C_2(v) = P \cup P_2(v) \cup P_k(v)$ is an odd hole of G for any $v \in V(G)$ using similar data structures in Case 1. If none of the $O(n)$ graphs $C_2(v)$ is an odd hole of G, then report the empty graph. Otherwise, report a shortest one of the graphs $C_2(v)$ that are odd holes.

The rest of the proof shows that the next choice of $j \in [2], k \in [3,5], x_0 \in M_G(C^*)$, and $\{x_1, \ldots, x_5\} \subseteq V(C^*)$ with $x_4x_5 \in E(C^*)$ for a shortest odd hole C^* of G yields a shortest odd hole $C_j(x_{3-j})$ of G: $M_G(C^*)$ is non-empty or else C^* is medium in G. Let B be a longest induced cycle of $G[C^* \cup M_G(C^*)]$ with $|V(B) \cap M_G(C^*)| = 1$. Let $B^* = B - x_0$ for the vertex $x_0 \in V(B) \cap M_G(C^*)$. By $M_G(C^*) \nsubseteq N_G(e)$ for any $e \in E(C^*)$ (or else C^* is medium in G), we

doi:10.6342/NTU202203481

have $\|B^*\| \geq 3$. Lemmas 2.9 and 2.10 imply an $x_4x_5 \in E(C^*)$ with

$$M_G(C^*) \subseteq N_G[\{x_0, x_4, x_5\}]. \tag{2.2}$$

Let $k = |V(B^*) \cap \{x_4, x_5\}| + 3$. Let B^* (respectively, $B^* - \{x_4, x_5\}$ be an x_1x_k-path (respectively, x_1x_3-path) such that an x_1x_5-path of C^* contains x_3 and x_4. Thus, $N_G(x_4x_5) \cap V(B^*) \subseteq \{x_3\}$ and x_1, x_3, x_4, and x_5 are in order in C^*. By maximality of $\|B\|$, we have

$$M_G(C^*) \subseteq (N_G(x_1) \cap N_G(x_k)) \cup N_G(\text{int}(B^*)). \tag{2.3}$$

Let $C^*(u, v)$ be the shortest uv-path of C^* for each $\{u, v\} \subseteq V(C^*)$. Let $j \in [2]$ such that $j = 1$ if and only if $\|B^*\| = \|C^*(x_1, x_k)\|$. B is a hole of G shorter than C^* by $x_0 \in M_G(C^*)$, so $\|B^*\|$ is even. Let x_2 be the interior vertex of the non-shortest x_1x_k-path of C^* with

$$\|C^*(x_1, x_2)\| = \|C^*(x_2, x_k)\| - j. \tag{2.4}$$

Thus, $C^*(x_j, x_k) \subseteq H$. By $x_0 \in M_G(C^*)$ and $\|B^*\| \geq 3$, we have $\|C^*(x_1, x_k)\| \geq 3$. By Equation (2.4),

$$C^* = C^*(x_1, x_k) \cup C^*(x_2, x_k) \cup C^*(x_1, x_2).$$

Based upon Lemma 2.6, we prove for either case of $j \in [2]$ that

$$C_j(x_{3-j}) = P \cup P_j(x_{3-j}) \cup P_k(x_{3-j})$$

is a shortest odd hole of G by ensuring the two statements below via the following immediate corollary of Lemmas 2.5 and 2.6: If the shortest uv-path $C^*(u, v)$ of a shortest odd hole C^* of G is contained by a subgraph H of the non-shallow and non-deep $G^* = G - M_G(C^*)$, then each shortest uv-path P of H is a shortest path of G^* and we call H a *witness* for P. See Figure 2.2.

doi:10.6342/NTU202203481

1. P is a shortest x_jx_k-path of G^*: By Equation (2.2), we have $C^*(x_j, x_k) \subseteq H \subseteq G^*$. H is a witness for P.

2. Each $P_i(x_{3-j})$ with $i \in \{j, k\}$ is a shortest x_ix_{3-j}-path of G^*: When $j = 1$, we have $B^* = C^*(x_1, x_k)$. By $\text{int}(B^*) \subseteq I$ and Equation (2.3), we have $C^*(x_i, x_2) \subseteq G_0 \subseteq G^*$ for each $i \in \{1, k\}$. G_0 is a witness for $P_1(x_2)$ and $P_k(x_2)$. When $j = 2$, we have $B^* = C^*(x_1, x_2) \cup C^*(x_2, x_k)$. By $V(C^*(x_1, x_2) - x_1) \subseteq I_1$ and $\text{int}(C^*(x_2, x_k)) \subseteq I$, we have $x_1 \in X_1$ and $\text{int}(B^*) \subseteq I_1 \cup I$. By Equation (2.3), we have $B^* \subseteq G_1$ and $V(G_1) \cap M_G(C^*) \subseteq X_1$, implying $C^*(x_1, x_i) \subseteq G_0(x_1) \subseteq G^*$ for each $i \in \{2, k\}$. $G_0(x_1)$ is a witness for $P_2(x_1)$ and $P_k(x_1)$. □

2.5.1 Proving Lemma 2.9

A path P of a shortest odd hole C of G is an *x-gap* (see, e. g., [24]) with $x \in M_G(C)$ if $G[P \cup \{x\}]$ is a hole of G (and thus $\|P\| \geq 2$). The shortestness of C implies that each x-gap is even.

Proof of Lemma 2.9. Let $x \in M_G(C) \setminus M^*_G(C)$, implying that $|N_G(x, C)| \leq 3$. Since $\|C\|$ is odd, there is an edge of C that is not contained by any x-gap, implying that $C[N_G(x, C)]$ contains exactly one edge of C. Since $x \in M_G(C)$, we have $|N_G(x, C)| = 3$ and thus $C = C(T)$ for a tripod T of G with $\|T_1\| = 1$. □

2.5.2 Proving Lemma 2.10

An $X \subseteq V(G)$ is *stable* if $E(G[X]) = \varnothing$. A $v \in V(G)$ (respectively, $uv \in E(G)$) is *X-complete* with $X \subseteq V(G)$ if $v \in N_G(x)$ (respectively, $\{u, v\} \subseteq N_G(x)$) holds for each $x \in X$.

Lemma 2.11. *For any stable $X \subseteq M^*_G(C)$ for a non-shallow shortest odd hole C of G, the number of X-complete edges of C is odd.*

Lemma 2.11 is stronger than [23, 5.1] in that G is allowed to be shallow. We first reduce Lemma 2.10 to Lemma 2.11 and then prove Lemma 2.11.

doi:10.6342/NTU202203481

Proof of Lemma 2.10. Assume for contradiction a G with minimum $|V(G)|$ violating the lemma. We have $M^*_G(C) = V(G) \setminus V(C)$. Let $x_0 \in M^*_G(C)$ with $M^*_G(C) \not\subseteq N_G(e) \cup N_G(x_0)$ for each $e \in E(C)$, which has to be anticomplete to $M^*_G(C) \setminus \{x_0\}$ by minimality of $|V(G)|$. Lemma 2.11 implies an edge x_1x_2 of $G[M^*_G(C)]$. The minimality of $|V(G)|$ implies for each $i \in [2]$ an edge $e_i \in E(C)$ that is adjacent to each vertex of $M^*_G(C) \setminus \{x_{3-i}\}$. Since Lemma 2.11 implies an $\{x_0, x_i\}$-complete edge f of C, $G[\{x_0, x_i\} \cup e_i]$ is not an induced x_0x_i-path P (with $\|P\| = 3$) or else $G[P \cup f]$ contains a 5-hole of G. Thus, each $i \in [2]$ admits an $\{x_0, x_i\}$-complete end v_i of e_i. By definition of x_0, each x_i with $i \in [2]$ is anticomplete to e_{3-i}. Hence, we have $v_1 \neq v_2$, implying $v_1v_2 \in E(C)$ or else $G[\{x_1, v_1, x_0, v_2, x_2\}]$ is a 5-hole. However, $e = v_1v_2$ is adjacent to each member of $M^*_G(C)$: if a $z \in M^*_G(C)$ is anticomplete to e, then $z \notin \{x_0, x_1, x_2\}$ and z is $\{e_1 - v_1, e_2 - v_2\}$-complete. Thus, $G[e_1 \cup e_2 \cup \{z\}]$ is a 5-hole, contradiction. □

Proof of Lemma 2.11. Assume for contradiction that an X with minimum $|X|$ violates the lemma, implying $|X| \geq 2$. A path P of C is an *xy-gap* with $\{x, y\} \subseteq X$ and $x \neq y$ if (i) P is an $\{x, y\}$-complete vertex (and thus $\|P\| = 0$) or (ii) P is a uv-path with $N_G(x, P) = \{u\}$ and $N_G(y, P) = \{v\}$ (and thus $\|P\| \geq 1$).

Claim A: There are vertices x and y of X such that C contains an odd xy-gap P and an even xy-gap Q.

We first reduce the lemma to Claim A and then prove Claim A. Observe that P and Q are disjoint or else $P \cup Q$ contains an odd x-gap, violating the shortestness of C. Thus, $|N_G(x, P \cup Q)| = |N_G(y, P \cup Q)| = 2$. If P and Q are not adjacent, then $G[P \cup Q \cup \{x, y\}]$ is an odd hole with length $\|C\|$ by shortestness of C. By $\{x, y\} \subseteq X$, the two vertices in $C - V(P \cup Q)$ are $\{x, y\}$-complete. We have $\|Q\| \neq 0$ or else $C[N_G(x)]$ is a 3-path, violating that G is non-shallow. Hence, $\|Q\| \geq 2$, implying an odd x-gap in $C[N_C[Q]]$, contradiction. Thus, P and Q are adjacent in C, implying that $R = C - V(P \cup Q)$ is an odd uv-path of C with $\min(|N_G(x, R)|, |N_G(y, R)|) \geq 2$.

doi:10.6342/NTU202203481

- If $\|R\| = 1$, then R is an $\{x, y\}$-complete edge of C. We have $\|Q\| \neq 0$ or else $C[N_G(x)]$ is a 3-path. By $\|Q\| \geq 2$, $C[N_C[Q]] - V(P)$ is an odd x-gap or y-gap, contradiction.
- If $\|R\| \geq 3$, then $S = R - \{u, v\}$ is a path of C disjoint and nonadjacent to $P \cup Q$. By the above observation, S contains no xy-gap, implying $N_G(x, S) = \varnothing$ or $N_G(y, S) = \varnothing$. If $N_G(x, S) = \varnothing$ (respectively, $N_G(y, S) = \varnothing$), then R is an odd x-gap (respectively, y-gap), contradiction.

Claim B: If each E_Y with $Y \in \mathscr{X} = 2^X \setminus \{\varnothing\}$ consists of the Y-complete edges of C, then there are an even number of edges in the set $F = \bigcup_{Y \in \mathscr{X}} E_Y$.

We reduce Claim A to Claim B. To see that P exists, let $U = N_G(X, C)$. Let $\mathscr{P}$ consist of the paths P with int$(P) \cap U = \varnothing$ and distinct ends in U. The paths in $\mathscr{P}$ are pairwise edge-disjoint. The union of the paths in $\mathscr{P}$ is C. By Claim B, there is an odd uv-path $P \in \mathscr{P}$ that is not an edge in F. There is an $\{x, y\} \subseteq X$ with $\{ux, vy\} \subseteq E(G)$. We have $x \neq y$ or else P is an odd x-gap of C, violating the shortestness of C. Thus, P is an odd xy-gap of C. To see that Q exists, assume for contradiction that all xy-gaps are odd. Thus, C contains no $\{x, y\}$-complete edge, since an $\{x, y\}$-complete vertex of C is an even xy-gap. Hence, C contains an even number of $\{x\}$-complete or $\{y\}$-complete edges. The number of edges of C contained by x-gaps or y-gaps is also even. Since an edge of C not contained by any xy-gaps has to be an $\{x\}$-complete or $\{y\}$-complete edge or contained by an x-gap or a y-gap, the number of edges in $Q = C - \text{int}(P)$ that are contained by xy-gaps is even. Therefore, the number of xy-gaps in Q is even, implying an $\{x, y\}$-complete end of Q, contradicting no even xy-gap in C. Claim A is proved.

It remains to prove Claim B. Observe that the minimality of $|X|$ implies that $|E_X|$ is even and $|E_Y|$ is odd for each $Y \in \mathscr{X} \setminus \{X\}$. Since $|\mathscr{X} \setminus \{X\}|$ is even, so is $\sum_{Y \in \mathscr{X}} |E_Y|$. Let each $X(e) \subseteq X$ with $e \in E(C)$ consist of the $V(e)$-complete vertices of X. For each $e \in E(C)$ and $Y \in \mathscr{X}$, we have $e \in E_Y$ if and only if $Y \subseteq X(e)$. Thus, each edge $e \in F$ belongs to exactly $2^{|X(e)|} - 1$ sets E_Y with $Y \in \mathscr{X}$. Therefore, $\sum_{e \in F} \left(2^{|X(e)|} - 1\right) = \sum_{Y \in \mathscr{X}} |E_Y|$ is even. Claim B is proved and so is the lemma. $\square$

doi:10.6342/NTU202203481

Chapter 3

Finding a Shortest Odd Hole

3.1 Technical Overview

Our $O(n^7)$-time algorithm to find an odd hole is almost one for finding a shortest odd hole. Among the four subroutines, only the one for (2) may find a non-shortest odd hole. Indeed, our $O(n^{13})$-time algorithm for finding a shortest odd hole is obtained by replacing our subroutine for (2) above with an $O(n^{13})$-time one for finding a shortest odd hole in a deep and non-shallow graph (Lemma 3.1), which improves upon Chudnovsky, Scott, and Seymour's $O(n^{14})$-time subroutine [23, 3.2] for finding a shortest odd hole in a graph containing "great pyramids", no "jewelled" shortest odd hole, and no 5-hole. Chudnovsky et al.'s subroutine enumerates all $O(n^{12})$ twelve-tuples $y = (y_0, \ldots, y_{11})$ of vertices and finds for each y in $O(n^2)$ time with the assistance of $(y_0, \ldots, y_4)$ a great pyramid H containing $\{y_5, \ldots, y_{11}\}$. Specifically, y_5 is the "apex" of H, $\{y_6, y_7, y_8\}$ forms the "base" of H (see §2.2), and $\{y_9, y_{10}, y_{11}\}$ consists of the interior marker (defined in §3.3) vertices of a path of H between its apex and base. Our improved $O(n^{13})$-time subroutine is based on a new observation (Claim 1 in the proof of Lemma 3.1, which strengthens [23, 7.2]) that a vertex in the base $\{y_6, y_7, y_8\}$ of H can be omitted in the enumeration, reducing the number of rounds from

doi:10.6342/NTU202203481

$O(n^{12})$ to $O(n^{11})$, without increasing the time $O(n^2)$ to pin down a shortest odd hole.

3.2 Proving Theorem 2

By Theorem 1, the rest of the chapter assumes without loss of generality that G contains odd holes and each odd hole of G has length at least 15, since all odd holes shorter than 15 can be listed in $O(m^3n^7)$ time.

Lemma 3.1. *It takes $O(m^3n^7)$ time to obtain a $C \subseteq G$ such that (1) C is a shortest odd hole of G or (2) G contains a shallow hole or no deep hole.*

Lemma 3.1 improves upon the $O(m^3n^8)$-time algorithm of [23, Lemma 3.2]. We first reduce Theorem 2 to Lemma 3.1 via Lemmas 2.2, 2.3, and 2.4 and then prove Lemma 3.1 in §3.3.

Proof of Theorem 2. Assume for contradiction that none of the four $C \subseteq G$ ensured by Lemmas 3.1, 2.2, 2.3, and 2.4 is a shortest odd hole of G. By Lemma 2.2, G is non-shallow. By Lemma 3.1, G is non-deep. By Lemma 2.3, G is non-medium, contradicting Lemma 2.4. Thus, it takes $O(m)$ time to obtain a shortest odd hole of G from the four C. □

3.3 Proving Lemma 3.1

A *c-trail* of a graph G for a $c = (c_0, \ldots, c_k)$ with $\{c_0, \ldots, c_k\} \subseteq V(G)$ is the union of one shortest $c_{i-1}c_i$-path of G per $i \in [k]$. We call $c = (c_0, \ldots, c_4)$ with $\{c_0, \ldots, c_4\} \subseteq V(P)$ a *marker* of a c_0c_4-path P of a deep graph G if $d_P(c_0, c_2) = \lceil \|P\|/2 \rceil$, $d_P(c_0, c_1) = \min(\delta_G, d_P(c_0, c_2))$, and $d_P(c_3, c_4) = \min(\delta_G, d_P(c_2, c_4))$.

Lemma 3.2 (Chudnovsky, Scott, and Seymour [23, 7.1])**.** *If $|N_G(x) \cap B(T)| \leq 1$ with $x \in M^*_G(C(T))$ holds for a tripod T of a deep G, then $G[N_G(x, T_1 \cup T_i)]$ is an edge for an $\{i, j\} = \{2, 3\}$ with $\|T_j\| \geq 3$.*

doi:10.6342/NTU202203481

Lemma 3.3 (Chudnovsky, Scott, and Seymour [23, 8.3, 8.4, and 8.5]). *Let T be a tripod of a deep and non-shallow G. Let $(a, b_2, b_3) = (a(T), T_2[B(T)], T_3[B(T)])$. For any marker $c = (a, c_1, c_2, c_3, b_i)$ of T_i with $\{i, j\} = \{2, 3\}$, the triple obtained from T by replacing T_i with a c-trail of the graph*

$$G - ((M^*_G(C(T)) \cup N_G[T_1 - a] \cup N_G[b_j]) \setminus \{a, c_1, c_2, c_3, b_i\})$$

remains a tripod of G.

We are ready to prove Lemma 3.1 by Lemmas 2.7, 2.9, 2.10, 2.11, 3.2, and 3.3.

Proof of Lemma 3.1. For any $c = (c_0, \ldots, c_k)$ with $\{c_0, \ldots, c_k\} \subseteq V(H)$ for a graph H, let $P_H(c_0, \ldots, c_k)$ be an arbitrary fixed c-trail, if any, of H. For each of the $O(m^2 n^7)$ choices of $\{x, a, c_1, c_2, c_3, d_1, d_2\} \subseteq V(G)$, $\{i, j\} = \{2, 3\}$, and $\{b, e\} \subseteq E(G)$ with $b = b_2 b_3$, spend $O(m)$ time to determine whether $G[P_2 \cup P_3]$ is an odd hole of G with

$$\begin{aligned}
Y &= (N_G(x) \cup N_G(e)) \setminus (V(e) \cup \{d_1, d_2\})\\
G_1 &= G - ((Y \cup (N_G[b] \setminus (N_G(b_2) \cap N_G(b_3)))) \setminus \{a, b_j\})\\
P_1 &= P_{G_1}(a, b_j) - b_j\\
G_i &= G - ((Y \cup N_G[P_1 - a] \cup N_G[b_j]) \setminus \{a, c_1, c_2, c_3, b_i\})\\
P_i &= P_{G_i}(a, c_1, c_2, c_3, b_i)\\
G_j &= G - (N_G[(P_1 \cup P_i) - a] \setminus \{a, b_j\})\\
P_j &= P_{G_j}(a, b_j).
\end{aligned}$$

If there are such odd holes of G, then report a shortest of them as the $O(m^3 n^7)$-time obtainable subgraph of G. Otherwise, report the empty graph. We prove that if G is deep and non-shallow, then the reported subgraph is a shortest odd hole of G by ensuring that (P_1, P_2, P_3) is a tripod of G for the following choice of $\{x, a, c_1, c_2, c_3, d_1, d_2\} \subseteq V(G)$, $\{b, e\} \subseteq E(G)$, and $\{i, j\} = \{2, 3\}$:

doi:10.6342/NTU202203481

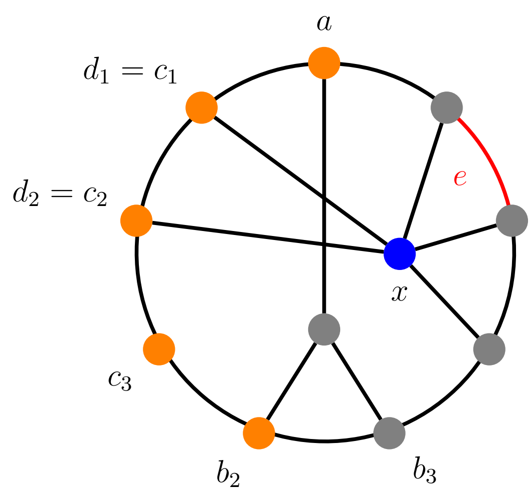


Figure 3.1: A choice of $\{x, a, c_1, c_2, c_3, d_1, d_2\} \subseteq V(G)$ and $\{b, e\} \subseteq E(C)$ with $(i, j) = (2, 3)$ in the proof of Lemma 3.1. The circle is a deep hole C with $\|C\| = 9$ of the deep graph G with $\delta_G = 1$. The orange vertices form a marker (a, c_1, c_2, c_3, b_2) of P_2. The edge e_2 in the proof of Claim 1 is c_1c_2. $I = \{c_1, c_2\}$.

Let T be a tripod of G. Let $C = C(T)$. For each $k \in \{2, 3\}$, let $b_k = T_k[B(T)]$ and $C_k = G[T_1 \cup T_k]$. Let $(a, b) = (a(T), b_2b_3)$. Thus, $a \notin N_G(b)$. We claim a choice of $\{x, d_1, d_2\} \subseteq V(G)$, $\{i, j\} = \{2, 3\}$, and $e \in E(C)$ satisfying the next two statements and choose $\{c_1, c_2, c_3\} \subseteq V(T_i)$ such that $c = (a, c_1, c_2, c_3, b_i)$ is a marker of T_i. See Figure 3.1.

Claim 1: $M^*_G(C) \subseteq Y$ (and hence P_1 is C-clean) and $Y \cap V(C_i - \{a, b_i\}) = \varnothing$.

Claim 2: Each $D_k = G[P_1 \cup T_k]$ with $k \in \{2, 3\}$ is a hole of G with $\|D_k\| \leq \|C_k\|$.

$G[V(T_1) \cup \{b_j\}] \subseteq G_1$ by Claim 1 with $T_1 \subseteq C_i$ and $V(T_1) \cap (N_G[b] \setminus (N_G(b_2) \cap N_G(b_3))) = \varnothing$. Thus,

$$1 \leq \|P_1\| \leq \delta_G \tag{3.1}$$

by $a \notin N_G(b)$. Let b_1 be the end of P_1 with $b_1b_j \subseteq P_{G_1}(a, b_j)$. $G[\{b_1, b_2, b_3\}]$ is a triangle or else $b_1 \in N_G[b]$ contradicts $b_1 \in V(G_1)$. By Claim 2 and Equation (3.1), (P_1, T_2, T_3) is a tripod of G, implying $\text{int}(T_i) \cap N_G[P_1 - a] = \varnothing$. Thus, $T_i \subseteq G_i$ and $\|P_i\| \leq \|T_i\|$. By Lemma 3.3, T_i is a c-trail

doi:10.6342/NTU202203481

of

$$G'_i = G - ((M^*_G(C) \cup N_G[P_1 - a] \cup N_G[b_j]) \setminus \{a, c_1, c_2, c_3, b_i\}).$$

By Claim 1, we have $T_i \subseteq G_i \subseteq G'_i$, implying that P_i is a c-trail of G'_i. By Lemma 3.3, the triple obtained from (P_1, T_2, T_3) by replacing T_i with P_i is a tripod of G, implying $\text{int}(T_j) \cap N_G[(P_1 \cup P_i) - a] = \varnothing$. Hence, we have $T_j \subseteq G_j$, implying $\|P_j\| \le \|T_j\|$. The definition of G_j implies $\text{int}(P_j) \cap N_G[(P_1 \cup P_i) - a] = \varnothing$. By $a \notin N_G(b)$, $D = G[P_i \cup P_j]$ is a hole of G with $\|D\| \le \|C\|$. If $\|D\| < \|C\|$, then a $G[P_1 \cup P_2 \cup P_3] - V(P_k - a)$ with $k \in [3]$ is an odd hole of G shorter than C by $1 \le \|P_k\| \le \|T_k\|$ for each $k \in [3]$, contradiction. Thus, D is a shortest odd hole of G, implying that (P_1, P_2, P_3) is a tripod of G.

The rest of the proof ensures Claims 1 and 2 in order. To see Claim 1, assume an $x \in M^*_G(C) \setminus N_G(b)$ or else the claim holds with $(x, e) = (b_2, b)$, $d_1 = T_1[B]$, $\{d_2\} = N_{T_2}(b_2)$, and $(i, j) = (2, 3)$. Lemma 2.10 implies an $e \in E(C)$ with minimum $\varphi(e) = |V(e) \cap \{a, b_2, b_3\}|$ such that $M^*_G(C)$ is contained by the set

$$N = (N_G(x) \cup N_G(e)) \setminus V(e).$$

Thus, $x \in N_G(e)$. By $x \notin N_G(b)$, we have $|N_G(x) \cap B(T)| \le 1$. Lemma 3.2 implies an $\{i, j\} = \{2, 3\}$ with $\|T_j\| \ge 3$ such that $G[N_G(x, C_i)]$ is an edge e_i. Assume for contradiction at least three vertices in the set

$$I = N \cap (V(C_i) \setminus \{a, b_i\}) = (V(e_i) \cup N_G(e, C_i)) \setminus \{a, b_i\}.$$

We have $e \notin E(T_i)$ or else $x \in N_G(e)$ implies $|I| \le 2$. By $x \in N_G(e) \setminus N_G(b)$, we have $e \ne b$, implying $e \in E(T_j)$. We have $V(e) \not\subseteq \text{int}(T_j)$ or else $N_G(e, C_i) \subseteq \{a\}$ implies $|I| \le 2$. Thus, $e = uv$ with $u \in \text{int}(T_j)$ and $v \in \{a, b_j\}$. If $v = a$, then $v \notin N_G(x)$ or else $|I| = |N_{C_i}(a)| = 2$. If $v = b_j$, then $v \notin N_G(x)$ by $x \notin N_G(b)$. By $\|T_j\| \ge 3$, the neighbor w of u in $T_j - v$ is in $\text{int}(T_j)$.

The minimality of $\varphi(e)$ implies a

$$y \in M^*_G(C) \setminus ((N_G(x) \cup N_G(uw)) \setminus \{u, w\}).$$

Thus, $P = xuvy$ is an induced 3-path of G. Lemma 2.11 implies an $\{x, y\}$-complete edge f of C. Thus $G[f \cup P]$ contains a 5-hole of G, contradiction. Hence, Claim 1 holds with $\{d_1, d_2\} = I$. See Figure 3.1.

It remains to prove Claim 2 by Equation (3.1) and Lemmas 2.7 and 2.9. We first ready Equations (3.2), (3.3), and (3.4) below and ensure that $V(P_1 - a)$ and $V(C)$ are disjoint. By Claim 1 and Lemma 2.7, an arbitrary (unnecessarily induced) uv-path P of $H = G[P_1 \cup C]$ with $\{u, v\} \subseteq V(C)$ and $\|P\| \leq \delta_G$ implies

$$d_C(u, v) \leq \|P\|. \tag{3.2}$$

Thus, $\|P\| \leq \delta_G$ and $\{u, v\} \subseteq V(T_k - a)$ with $\{k, \ell\} \in \{2, 3\}$ imply $d_C(u, v) = T_k[u, v]$ by $d_C(u, v) \leq \delta_G < \|T_\ell\|$. Observe that $\|P_1\| < \delta_G$ implies contradiction from $\|T_2\| \leq \|P_1\| + 1 \leq \delta_G < \|T_2\|$ by Equation (3.2) via the ab_2-path $P_1 \cup b_1b_2$ of H with length at most δ_G. Therefore, by Equation (3.1), we have

$$\|P_1\| = \delta_G. \tag{3.3}$$

Hence, each $\|P_1\| + \|T_k\|$ with $k \in \{2, 3\}$ is odd. Also, $V(P_1 - a)$ and $V(C)$ are disjoint or else a vertex $v \in V(P_1 - a) \cap V(T_k)$ with $\{k, \ell\} = \{2, 3\}$ leads to contradiction from

$$\|T_k\| + 1 = \|T_k \cup b_k b_\ell\| = d_C(a, v) + d_C(v, b_\ell) \leq \|P_1[a, v]\| + \|P_1[v, b_1] \cup b_1 b_\ell\| = \|P_1\| + 1 \leq \|T_k\|$$

by Equations (3.1) and (3.2) via the av-path $P_1[a, v]$ and the vb_ℓ-path $P_1[v, b_1] \cup b_1 b_\ell$ of H. As a result, if D_k with $k \in \{2, 3\}$ is not a hole of G, then there is an edge $u_1 u_k \in E(D_k) \setminus \{b_1 b_k\}$ with $u_1 \in V(P_1 - a)$ and $u_k \in V(T_k - a)$. Moreover, each edge $u_1 u_k \in E(D_k)$ with $u_1 \in \text{int}(P_1)$ and

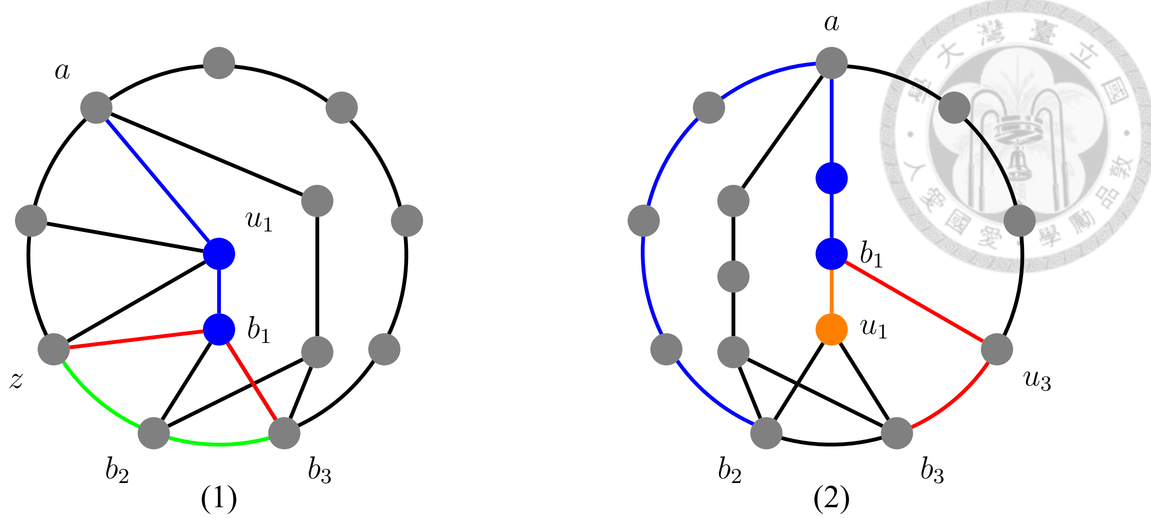


Figure 3.2: Illustrating Claim 3.2 in the proof of Lemma 3.1. The circle in each example is a deep hole C with $\|C\| = 9$ of the graph G. (1) An illustration with $k = 2$ and $\delta_G = 2$ showing that b_1 is anticomplete to $C - \{a, b_2, b_3\}$. The blue path denotes the path $\|P_1\|$. The graph obtained from C by replacing the green path with the red path is a shortest odd hole C^* of G. $G[N_G(u_1, C^*)]$ is the 3-path $T_k[a, z] \cup zb_1$. (2) An illustration with $k = 3$ and $\delta_G = 3$ for the case $P[a, u_1]+1 = T_k[a, u_k]$ in the proof of Lemma 3.1. The orange, red, and blue paths denote Q_1, Q_2, and Q_3, respectively. (Q_1, Q_2, Q_3) satisfies Conditions Z.

$u_k \in V(T_k - a)$ satisfies

$$\|P_1[a, u_1]\| \leq \|T_k[a, u_k]\| \leq \|P_1[a, u_1]\| + 1 : \tag{3.4}$$

If $au_1 \in E(P_1)$, then $\|P_1[a, u_1]\| = 1 \leq \|T_k[a, u_k]\|$. Otherwise, Equation (3.2) via the u_kb_ℓ-path $u_ku_1 \cup P_1[u_1, b_1] \cup b_1b_\ell$ of H implies $\|T_k[u_k, b_k]\| + 1 \leq \|P_1[u_1, b_1]\| + 2$. Hence,

$$\|P_1[a, u_1]\| = \|P_1\| - \|P_1[u_1, b_1]\| \leq \|T_k\| - 1 - \|T_k[u_k, b_k]\| + 1 = \|T_k[a, u_k]\|.$$

Equation (3.2) via the au_k-path $P_1[a, u_1] \cup u_1u_k$ of H implies $\|T_k[a, u_k]\| \leq \|P_1[a, u_1]\| + 1$.

To prove Claim 2 by contradiction, assume that D_2 or D_3 is not a hole of G. We first show that b_1 is anticomplete to $C - \{a, b_2, b_3\}$. Suppose that b_1 is adjacent to an $\text{int}(T_k)$ with $k \in \{2, 3\}$.

doi:10.6342/NTU202203481

Since $\delta_G = 1$ implies $b_1 \in M^*_G(C)$ (violating Claim 1 for P_1 being C-clean), we have $\delta_G \geq 2$. Lemma 2.9 with $\delta_G \geq 2$ implies $M^*_G(C) = M_G(C)$. Thus, b_1 is adjacent to exactly one vertex $z \in \text{int}(T_k)$ and $C[\{z\} \cup b_2b_3]$ is the 2-path zb_kb_ℓ, implying that $C^* = G[(C - b_k) \cup \{b_1\}]$ is also a shortest odd hole of G. By Lemma 2.7 via the ab_1-path P_1 with length $\delta_G < d_{C^*}(a, b_1)$, the C-clean path P_1 is not C^*-clean. Thus, the neighbor u_1 of b_1 in P_1 is a vertex of $M^*_G(C^*)$. Equation (3.4) implies $|N_G(u_1, T_k - a)| \leq 2$, so P_1 is the 2-path au_1b_1 and the vertex u_k of T_k with $\|P_1[a, u_1]\| + 1 = \|T_k[a, u_k]\|$ is a neighbor of u_1. Since $\|P_1\| + \|T_k\|$ is odd by Equation (3.3), we have $\|T_k\| = \|P_1\| + 1$ (implying $u_k = z$) or else $G[u_1b_1 \cup T_k[u_k, z]]$ is an odd hole of G shorter than C. As a result, $G[N_G(u_1, C^*)]$ is the 3-path $T_k[a, z] \cup zb_1$, contradicting that G is non-shallow. See Figure 3.2(1). Having shown that b_1 is anticomplete to $C - \{a, b_2, b_3\}$, we know a $u_1u_k \in E(D_k)$ with $\{k, \ell\} = \{2, 3\}$, $u_1 \in \text{int}(P_1)$, and $u_k \in V(T_k - a)$ that minimizes $\phi(u_1, u_k) = n \cdot d_{P_1}(u_1, b_1) + d_{T_k}(u_k, b_k)$. Since $\|P_1\| + \|T_k\|$ is odd by Equation (3.3), we have $\|T_k[a, u_k]\| = \|P_1[a, u_1]\| + 1$ by Equation (3.4) or else $G[P_1[u_1, b_1] \cup T_k[u_k, b_k]]$ is an odd hole of G shorter than C. Let $(Q_1, Q_2) = (P_1[u_1, b_1], u_1u_k \cup T_k[u_k, b_k])$. We have $1 \leq \|Q_1\| < \|Q_2\|$. Let Q_3 be a shortest u_1b_ℓ-path of $G[P_1[u_1, a] \cup T_\ell]$. Equation (3.4) implies $\|Q_1\| < \|Q_3\|$ and that $Q_2 - u_k$ is anticomplete to $\text{int}(Q_3)$. By minimality of $\phi(u_1, u_k)$, $Q_1 - u_1$ is anticomplete to $\text{int}(Q_3)$. Conditions Z hold for (Q_1, Q_2, Q_3) or (Q_1, Q_3, Q_2). Thus, $\|T_1\| > \|Q_1\| \geq \delta_G = \|T_1\|$, contradiction. See Figure 3.2(2). $\square$

doi:10.6342/NTU202203481

Chapter 4

Finding a Non-shortest Induced Path

4.1 Technical Overview

A subroutine B taking an ℓ-tuple of $V(G)$ as the only argument is a uv-*trailblazer of degree* ℓ for G if running B on all ℓ-tuples of $V(G)$ always reports a uv-trail of G unless G is uv-trailless. We call an ℓ-tuple of $V(G)$ on which B reports a uv-trail of G a *trail marker* for B. An $O(f(n))$-time uv-trailblazer of degree ℓ for G immediately implies the following $O(n^\ell \cdot f(n))$-time *trailblazing algorithm* for G: Run B on each ℓ-tuple $(a_1, \ldots, a_\ell)$ of $V(G)$ to either obtain a uv-trail of G or

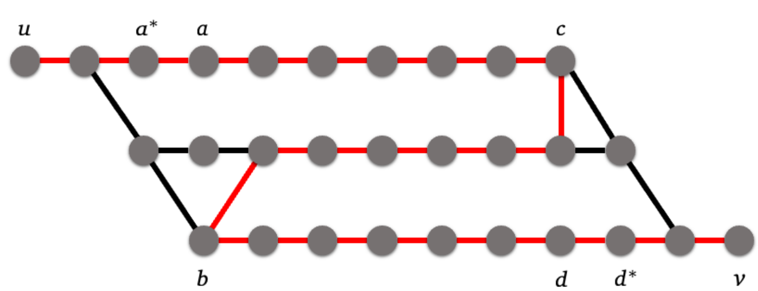


Figure 4.1: The red uv-path P is the only uv-trail of the uv-straight graph G. The twist pair of P is (c, b). The twist of P is 6. $P[a^*, c]$ and $P[b, d^*]$ form a pair of wings for the quadruple (a, b, c, d) of $V(G)$ in G.

doi:10.6342/NTU202203481

ensure that $(a_1, \ldots, a_\ell)$ is not a trail marker for B. If none of the $O(n^\ell)$ iterations produces a uv-trail of G, then report that G is uv-trailless.

A graph H is *uv-straight* [7] if $\{u, v\} \subseteq V(H)$ and each vertex of H belongs to at least one shortest uv-path of H. For instance, the graph in Figure 4.1 is uv-straight. Berger et al.'s algorithm starts with an $O(n^3)$-time preprocessing step (see Lemma 4.1) that either reports a uv-trail of G or obtains a uv-straight graph H with $V(H) \subseteq V(G)$ such that (a) a uv-trail of G can be obtained from a uv-trail of H in $O(n^2)$ time and (b) if H is uv-trailless, then so is G. If no uv-trail is reported by the preprocessing, then the main procedure runs an $O(n^{18})$-time trailblazing algorithm on the uv-straight graph H based on an $O(n^4)$-time degree-14 uv-trailblazer for H. As for postprocessing, if a uv-trail of H is obtained by the main procedure, then report a uv-trail of G obtainable in $O(n^2)$ time as ensured by the preprocessing. Otherwise, report that G is uv-trailless.

Our $O(n^{4.75})$-time algorithm adopts the preprocessing and postprocessing steps of Berger et al., while reducing the preprocessing time from $O(n^3)$ to $O(n^\omega)$ (see Lemma 4.5). For the benefit of the main procedure, we run a second preprocessing step, taking $O(n^{4.75})$ time via the witness matrix of Galil and Margalit [48], to compute a static data structure from which a pair of "wings" that are some disjoint paths in H, if any, for each quadruple of $V(H)$ can be obtained in $O(n)$ time (see Lemma 4.6). Our main procedure is also a trailblazing algorithm, based on a faster uv-trailblazer of a much lower degree for H: We reduce the time from $O(n^4)$ to $O(n^2 \log^2 n)$ and largely bring down the degree from 14 to 2. Thus, the main procedure runs in $O(n^4 \cdot \log^2 n)$ time, even faster than the second preprocessing step.

The key to our improved uv-trailblazer is a new observation, described by Lemma 4.4, on any shortest uv-trail P of a uv-straight graph G. Specifically, Berger et al.'s algorithm looks for a uv-trail in G that consists of (1) a shortest us-path S of G containing 7 guessed vertices and a shortest tv-path T of G containing another 7 guessed vertices such that S and T are anticomplete in G and (2) a shortest st-path Q of $G_{S,T} = G - (N_G[S \cup T - \{s, t\}] \setminus \{s, t\})$. Lemma 4.4

doi:10.6342/NTU202203481

ensures that much fewer guessed vertices on S and T suffice to guarantee that Q stays intact in $G_{S,T}$. To illustrate the usefulness of Lemma 4.4, we show in §4.2 that three lemmas of Berger et al. [7] (i. e., Lemmas 4.1, 4.2, and 4.3) together with Lemma 4.4 already yield an $O(n^2)$-time uv-trailblazer of degree 4 for G, leading to a simple $O(n^6)$-time trailblazing algorithm on G. More precisely, if a and b (respectively, c and d) are the vertices that are farthest apart from each other in P having the minimum identical distance to u (respectively, v) in G, then (a, b, c, d) is a trail marker for an $O(n^2)$-time uv-trailblazer for G: Due to the symmetry between u and v in G, Lemma 4.4 guarantees an $O(n^2)$-time obtainable uv-trail of G that contains the precomputed pair of "wings" for this (a, b, c, d).

Our proof of Theorem 3 in §4.3 further displays the usefulness of Lemma 4.4. We show that the aforementioned vertices a and b in P actually form a trail marker (a, b) for an $O(n^2 \log^2 n)$-time uv-trailblazer for G. Dropping both c and d from the trail marker (a, b, c, d) of §4.2 inevitably increases the time of the uv-trailblazer for G. We manage to keep the time of a degree-two uv-trailblazer as low as $O(n^2 \log^2 n)$ via the dynamic data structure of Holm, de Lichtenberg, and Thorup [56] supporting efficient edge updates and connectivity queries for G (see Lemma 4.7). To make our proof of Theorem 3 in §4 self-contained, a simplified proof of Lemma 4.3 is included in §4.2. Since Lemmas 4.1 and 4.2 are implied by Lemmas 4.5 and 4.6, which are proved in §4.3, our proof for the $O(n^6)$-time algorithm in §4.2 is also self-contained.

4.2 A simpler algorithm

Let $\{u, v\} \subseteq V(G)$. Let $h(x) = d_G(u, x)$ be the *height* of a vertex x in G. If xy is an edge of G, then $|h(x) - h(y)| \leq 1$.

Lemma 4.1 (Berger et al. [7, Lemma 2.2]). *For any vertices u and v of an n-vertex connected graph G, it takes $O(n^3)$ time to obtain (1) a uv-trail of G or (2) a uv-straight graph H with $V(H) \subseteq V(G)$*

such that (a) a uv-trail of G is $O(n^2)$-time obtainable from that of H and (b) if H is uv-trailless, then so is G.

A path of G is *monotone* [7] if all of its vertices have distinct heights in G. A monotone xy-path of G is a shortest xy-path of G. The converse may not hold. A shortest xy-path of G with $\{x, y\} \cap \{u, v\} \neq \varnothing$ is monotone. A monotone a^*c-path W_1 of G containing a vertex a and a monotone bd^*-path W_2 of G containing a vertex d with

$$h(a^*) + 1 = h(a) = h(b) \leq h(c) = h(d) = h(d^*) - 1$$

form a pair (W_1, W_2) of *wings* for the quadruple (a, b, c, d) of $V(G)$ in G if

$$d_{G[W_1 \cup W_2]}(a^*, d^*) > \|W_1\| + \|W_2\|,$$

that is, $W_1 - c$ (respectively, W_1) and W_2 (respectively, $W_2 - b$) are anticomplete in G. An (a, b, c, d) is *winged* in G if G admits a pair of wings for (a, b, c, d). See also Figure 4.1 for an example.

Lemma 4.2 (Berger et al. [7, Lemma 2.1])**.** *It takes $O(n^6)$ time to compute a data structure from which the following statements hold for any quadruple (a, b, c, d) of $V(G)$ for an n-vertex graph G:*

1. *It takes $O(1)$ time to determine whether (a, b, c, d) is winged in G.*
2. *If (a, b, c, d) is winged in G, then it takes $O(n)$ time to obtain a pair of wings for (a, b, c, d) in G.*

We comment that Lemma 2.1 of Berger et al. [7] is slightly different from Lemma 4.2, but their proof is easily adjustable into one for Lemma 4.2. See also §4.3 for a proof of Lemma 4.6, which implies and improves upon Lemma 4.2.

Let G be a uv-straight graph. If $h(s) - h(t)$ is maximized by the vertices s and t of a uv-path P of

doi:10.6342/NTU202203481

G such that $P[u, s]$ is a shortest us-path of G and $P[t, v]$ is a shortest tv-path of G, then the *twist* [7] of P is $h(s) - h(t)$ and we call (s, t) the *twist pair* of P. See also Figure 4.1 for an example. If (s, t) is the twist pair of a uv-path P of G, then $P[u, s]$ and $P[t, v]$ are disjoint if and only if P is a non-shortest uv-path of G. The next lemma is also needed in §4.3. To make our proof of Theorem 3 in §4.3 self-contained, we include a proof of Lemma 4.3 simplified from that of Berger et al. [7, Lemma 2.3].

Lemma 4.3 (Berger et al. [7, Lemma 2.3])**.** *If (s, t) is the twist pair of a shortest uv-trail P of a uv-straight graph G, then $h(s) \geq h(x) \geq h(t)$ holds for each vertex x of $P[s, t]$.*

Proof. Let $I = V(P[s, t]) \setminus \{s, t\}$. Let s^* (respectively, t^*) be the neighbor of s (respectively, t) in $P[s, t]$. By definition of (s, t), we have $h(s^*) \leq h(s)$ and $h(t^*) \geq h(t)$. If $I = \varnothing$, then $(s^*, t^*) = (t, s)$ implies the lemma. Otherwise, it suffices to prove $h(s) \geq h(x) \geq h(t)$ for each $x \in I$. If $h(x) > h(s)$ were true for the $x \in I$ maximizing the lexicographical order of $(h(x), d_{P[s,t]}(x, t))$, then the concatenation of $P[u, x]$ and a shortest xv-path of G is a uv-trail (containing s^*) of G shorter than P. If $h(x) < h(t)$ were true for the $x \in I$ minimizing the lexicographical order of $(h(x), d_{P[s,t]}(x, t))$, then the concatenation of a shortest ux-path of G and $P[x, v]$ is a uv-trail (containing t^*) of G shorter than P. □

A monotone uc-path S of G with $h(c) = h(s)$ is a *sidetrack* for a uv-trail P of G with twist pair (s, t) if satisfying the following *Conditions T*.

T1: $d_{G[S \cup T]}(u, v) > \|S\| + \|T\|$ holds for a monotone tv-path T of G.

T2: The vertex a of S with $h(a) = h(t)$ is on the monotone subpath $P[u, s]$.

The inequality of Condition T1 is equivalent to the statement that $S - c$ (respectively, S) and T (respectively, $T - t$) are anticomplete in G. Thus, $S[a^*, c]$ and $T[t, d^*]$ form a pair of wings for (a, t, c, d) in G, where a^* is the vertex of S with $h(a^*) = h(a) - 1$ and dd^* is the edge of T with $h(s) = h(d) = h(d^*) - 1$. See Figure 4.2 for an example. The key to our largely improved uv-

doi:10.6342/NTU202203481

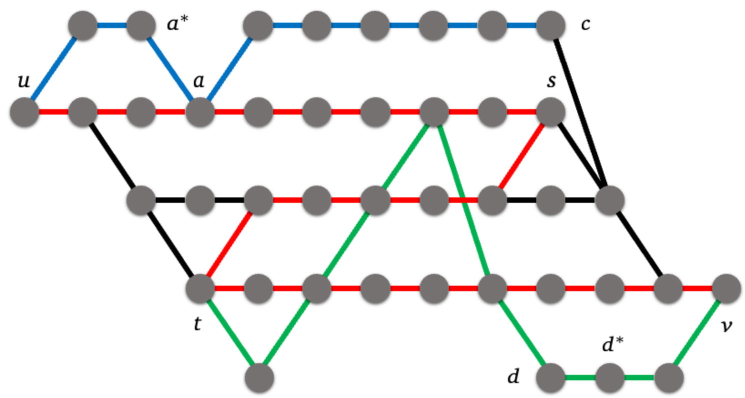


Figure 4.2: The blue uc-path is a sidetrack S for the red uv-trail P of the uv-straight graph G. Each of $P[t, v]$ and the green tv-path can be a monotone tv-path T satisfying Condition T1.

trailblazers in §4.2 and §4.3 is the following lemma, whose proof is illustrated in Figure 4.3.

Lemma 4.4. *If S is a sidetrack for a shortest uv-trail P of a uv-straight graph G with twist pair (s, t), then*

$$d_{G[S\cup P[s,t]]}(u, t) \geq d_P(u, t).$$

Proof. Condition T1 implies a monotone tv-path T of G with $d_{G[S\cup T]}(u, v) > \|S\| + \|T\|$. Assume for contradiction a shortest ut-path Q of $G[S \cup P[s, t]]$ with $\|Q\| < d_P(u, t)$, implying $d_{G[Q\cup T]}(u, v) < \|P\|$. By $t \notin V(S)$, Q contains an edge xy with $x \in V(S)$ and $y \in V(P[s, t])$ that minimizes $d_{P[s,t]}(y, t)$. Let R be a shortest uv-path of $G[Q \cup T]$. If x were not in $V(R)$, then $N_G(S[u, x] - x) \cap V(T) \neq \varnothing$, violating Condition T1. Hence, R contains x and, thus, y. Since R is an induced uv-path of G with $\|R\| < \|P\|$, we have $\|R\| = h(v)$, implying that R is monotone. By $d_R(u, x) < d_R(u, y)$,

$$h(x) + 1 = h(y). \tag{4.1}$$

By $\|Q\| + \|P[t, v]\| < \|P\|$, the concatenation of Q and $P[t, v]$ is a non-induced uv-path of G, implying that $G[Q \cup P[t, v]]$ contains a monotone uv-path R'. Let $x'y'$ be the edge of R' with

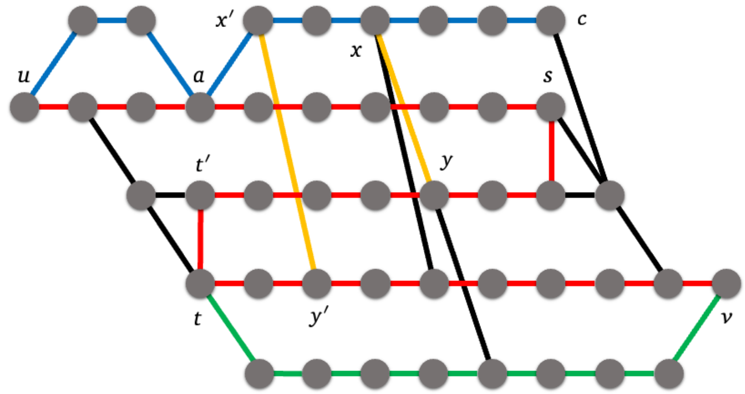


Figure 4.3: An illustration for the proof of Lemma 4.4. The red path denotes a shortest uv-trail P of the uv-straight graph G. The blue monotone path denotes a sidetrack S for P. The green path denotes a monotone path T satisfying Condition T1.

$x' \in V(S) \cap V(Q)$ and $y' \in V(P[t,v])$ that maximizes $h(y')$. By $d_{R'}(u,x') < d_{R'}(u,y')$,

$$h(x') + 1 = h(y'). \tag{4.2}$$

We know $h(x') \neq h(t) - 1$ or else $y' = t$ violates Condition T1. We know $h(x') \neq h(t)$ or else Condition T2 violates that P is induced. By $h(x') \geq h(t) + 1$ and Equation (4.2), y' and t are anticomplete in G. Let t' be the vertex closest to y in $P[y,t]$ with $h(t') = h(t)$, implying that y' and t' are anticomplete in G no matter whether $t' = t$ or not. By $h(x) \geq h(x') \geq h(t) + 1$ and Lemma 4.3, the concatenation P' of a shortest ut'-path of G, $P[t',y]$, the edge yx, and a shortest xv-path of $G[S[x',x] \cup P[y',v]]$ is an induced uv-path of G shorter than P. By Equation (4.1) and $d_{P'}(u,x') < d_{P'}(u,y')$, we have that P' is a uv-trail of G, contradicting the definition of P. $\square$

We are ready to describe and justify an $O(n^6)$-time algorithm that either reports a uv-trail of G or ensures that G is uv-trailless.

Our $O(n^6)$-time algorithm Apply Lemma 4.1 in $O(n^3)$ time to either report a uv-trail of G as stated in Lemma 4.1(1) or make G a uv-straight graph satisfying Conditions (a) and (b) of Lemma 4.1(2) with respect to the original G. If no uv-trail is reported in the previous step, then apply Lemma 4.2 to obtain the data structure D for the winged quadruples of G in $O(n^6)$ time. With the standard $O(n^2)$-time postprocessing readied by the preprocessing, it remains to show an $O(n^2)$-time degree-4 uv-trailblazer for the uv-straight graph G, which immediately leads to an $O(n^6)$-time trailblazing algorithm that either reports a uv-trail of G or ensures that G is uv-trailless.

Let B be the following $O(n^2)$-time subroutine, taking a quadruple (a, b, c, d) of $V(G)$ as the argument: Determine in $O(1)$ time from the data structure D whether (a, b, c, d) is winged in G. If not, then exit. Otherwise, obtain in $O(n)$ time from D a pair (W_1, W_2) of wings for (a, b, c, d) in G. Since G is uv-straight, it takes $O(n^2)$ time to obtain a monotone uc-path S of G containing W_1 and a monotone bv-path T of G containing W_2. Obtain in $O(n^2)$ time the subgraph $G_{c,b}$ of G induced by

$$\{x \in V(G) : h(b) \leq h(x) \leq h(c)\} \setminus ((N_G[S - c] \cup N_G[T - b]) \setminus \{c, b\}).$$

If c and b are not connected in $G_{c,b}$, then exit. Otherwise, report the concatenation $P_{c,b}$ of (i) the uc-path S, (ii) a shortest cb-path of $G_{c,b}$, and (iii) the bv-path T.

By definition of S, T, and $G_{c,b}$, the uv-path $P_{c,b}$ of G reported by $B(a, b, c, d)$ is induced in G, which is not monotone by $h(b) \leq h(c)$. Thus, $P_{c,b}$ is a uv-trail of G.

Let P be an arbitrary unknown shortest uv-trail of G with twist pair (s, t). Let a (respectively, d) be the vertex of the monotone $P[u, s]$ (respectively, $P[t, v]$) with $h(a) = h(t)$ (respectively, $h(d) = h(s)$). See Figure 4.4 for an illustration. The rest of the section shows that (a, t, s, d) is a trail marker for B.

Observe that $P[a^*, s]$ and $P[t, d^*]$ with the neighbor a^* of a in $P[u, a]$ and the neighbor d^* of d in $P[d, v]$ form a pair of wings for (a, t, s, d) in G. Thus, the quadruple (a, t, s, d) is winged in

doi:10.6342/NTU202203481

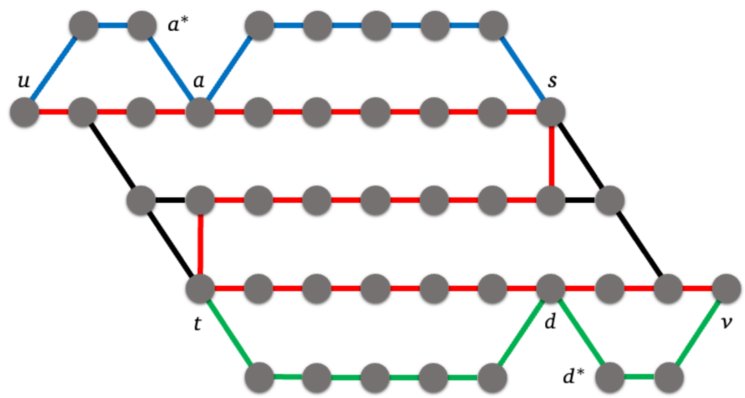


Figure 4.4: An illustration for the proof that B is a uv-trailblazer of degree four. The red path denotes a shortest uv-trail of the uv-straight graph G. The blue and green paths denote a monotone us-path and a monotone tv-path of G containing a precomputed pair of wings for (a, t, s, d) that need not coincide with P except at a, t, s, and d.

G. The monotone us-path S of G containing W_1 is a sidetrack for P, since the monotone tv-path T of G containing W_2 satisfies Conditions T1 and T2 for S. Due to the symmetry between u and v in G, the monotone vt-path T of the vu-straight graph G is also a sidetrack for the shortest vu-trail P of G with twist pair (t, s), since the monotone su-path S of G satisfies Conditions T1 and T2 for T. Lemma 4.3 guarantees $h(t) \leq h(x) \leq h(s)$ for each vertex x of $P[s, t]$. By Lemma 4.4, $P[s, t] - \{s, t\}$ is anticomplete to both $S - s$ and $T - t$, implying that $P[s, t]$ is a path of $G_{s,t}$. Since s and t are connected in $G_{s,t}$, the subroutine call $B(a, t, s, d)$ outputs a uv-trail $P_{s,t}$ of G in $O(n^2)$ time. Hence, (a, t, s, d) is indeed a trail marker of B.

As a matter of fact, $P_{s,t}$ is a shortest uv-trail of G due to $\|P_{s,t}\| = \|P\|$. Since the preprocessing and postprocessing may ruin the shortestness of the reported uv-trail, we have an $O(n^6)$-time algorithm on an n-vertex general (respectively, uv-straight) graph G that either reports a general (respectively, shortest) uv-trail of G or ensures that G is uv-trailless.

doi:10.6342/NTU202203481

4.3 Proof of Theorem 3

This section gives a self-contained proof of Theorem 3. The *product* of $m \times m$ Boolean matrices A and B is the $m \times m$ Boolean matrix C such that $C(i,k) =$ *true* if and only if $A(i,j) = B(j,k) =$ *true* holds for an index j. The following lemma implies and improves upon Lemma 4.1, which takes $O(n^3)$ time to obtain a uv-trail of G from a uv-trail of H.

Lemma 4.5. *For any vertices u and v of an n-vertex connected graph G, it takes $O(n^\omega)$ time to obtain (1) a uv-trail of G or (2) a uv-straight graph H with $V(H) \subseteq V(G)$ such that (a) a uv-trail of G can be obtained from a uv-trail of H in $O(n^2)$ time and (b) if H is uv-trailless, then so is G.*

Proof. We adopt the proof of Berger et al. [7, Lemma 2.2] with slight simplification and improvement. It takes $O(n^2)$ time to obtain the maximal set $F \subseteq V(G)$ such that $G[F]$ is uv-straight. If $F = V(G)$, then the lemma is proved by returning $H = G$. The rest of the proof assumes $F \subsetneq V(G)$. It takes $O(n^\omega)$ time to determine whether some connected component K of $G - F$ admits nonadjacent vertices x and y of $N_G(K) \subseteq F$ with $h(x) < h(y)$. If there is such a (K, x, y), then a shortest uv-path of $G[P_x \cup K \cup P_y]$ for any shortest ux-path P_x and yv-path P_y of G is a uv-trail of G obtainable in $O(n^2)$ time, proving the lemma. Otherwise, let H be the union of the uv-straight $G[F]$ and the $O(n^\omega)$-time obtainable graph H' with $V(H') = F$ (via contracting each connected component of $G - F$ into a single vertex and then squaring the adjacency matrix) such that distinct vertices x and y are adjacent in H' if and only if $\{x, y\} \subseteq N_G(K)$ holds for a connected component K of $G - F$. Observe that each edge xy of H' with $h(x) \neq h(y)$ is also an edge of $G[F]$. By $|h(x) - h(y)| \leq 1$ for all edges xy of H', H remains uv-straight and $d_H(u, x) = h(x)$ holds for each $x \in F$. To see Condition (a), for any given uv-trail Q of H, let P be an $O(n^2)$-time obtainable non-monotone uv-path of G obtained from Q by replacing each edge xy of Q not in $G[F]$ with a shortest xy-path P_{xy} of $G - (F \setminus \{x, y\})$. If P were not induced, then there is an edge zz' of $G[P]$ not in P with $z \in V(P_{xy})$ and $z' \in V(P_{x'y'})$ for distinct edges xy and $x'y'$

doi:10.6342/NTU202203481

of Q that are not in $G[F]$. Thus, $\{x, y, x', y'\} \subseteq N_G(K)$ holds for some connected component K of $G - F$. By definition of H', $H[\{x, y, x', y'\}]$ is complete, contradicting that Q is an induced path of H. Thus, P is a uv-trail of G, proving Condition (a). As for Condition (b), let P be a uv-trail of G. For any distinct vertices x and y of P such that $P[x, y]$ is a maximal subpath of P contained by $G[\{x, y\} \cup K]$ for some connected component K of $G - F$, $P[x, y]$ is an induced xy-path of $G[\{x, y\} \cup K]$. The path Q obtained from P by replacing each such $P[x, y]$ by the edge xy of H' is an induced uv-path of H. If Q were a shortest uv-path of H, then $|h(x) - h(y)| = 1$ holds for each edge xy of Q, implying that each edge xy of Q is an edge of P, contradicting that P is a uv-trail of G. $\square$

The bottleneck of our algorithm for Theorem 3 comes from the following lemma, which implies and improves upon Lemma 4.2 that takes $O(n^6)$ time.

Lemma 4.6. *It takes $\tilde{O}(n^{2\omega})$ time to compute a data structure from which the following statements hold for any quadruple (a, b, c, d) of $V(G)$ for an n-vertex graph G:*

1. *It takes $O(1)$ time to determine whether (a, b, c, d) is winged in G.*
2. *If (a, b, c, d) is winged in G, then it takes $O(n)$ time to obtain a pair of wings for (a, b, c, d) in G.*

Proof. The lemma holds clearly for the quadruples (a, b, c, d) of $V(G)$ with $h(c) \leq h(a) + 1$. The rest of the proof handles those with $h(a) + 2 \leq h(c)$. A pair of wings for such an (a, b, c, d) must be anticomplete in G. It takes $O(n^4)$ time to obtain the $n^2 \times n^2$ Boolean matrix A such that $A((a, b), (c, d)) =$ *true* if and only if (i) $h(a) = h(b) \leq h(c) = h(d) \leq h(a) + 1$ and (ii) G admits a pair of anticomplete wings for (a, b, c, d). The transitive closure $C = A^n$ of A can be obtained in $O(n^{2\omega} \cdot \log n)$ time via obtaining A^{2^i} in the i-th iteration. That is, for each (a, b, c, d), we have $C((a, b), (c, d)) =$ *true* if and only if (i) $h(a) = h(b) \leq h(c) = h(d)$ and (ii) G admits a pair of anticomplete wings for (a, b, c, d) in G. Statement 1 is proved. Statement 2 is immediate

doi:10.6342/NTU202203481

from the $\tilde{O}(n^{2\omega})$-time obtainable $n^2 \times n^2$ witness matrix W for C by, e. g., Galil and Margalit [48]: if $C((a,b),(c,d)) =$ *true* and $h(a)+2 \leq h(c)$, then $W((a,b),(c,d))$ is a vertex pair (x,y) with $h(a) < h(x) < h(c)$ and $C((a,b),(x,y)) = C((x,y),(c,d)) =$ *true*. □

The following dynamic data structure for a graph supports efficient edge updates and connectivity queries.

Lemma 4.7 (Holm, de Lichtenberg, and Thorup [56]). *There is a data structure for an initially empty n-vertex graph that supports each edge insertion and edge deletion in amortized $O(\log^2 n)$ time and answers whether two vertices are connected in $O(\log n/\log\log n)$ time.*

We are ready to prove Theorem 3.

Our $O(n^{4.75})$-time algorithm Apply Lemma 4.5 in $O(n^\omega)$ time to either report a uv-trail of G as in Lemma 4.5(1) or make G a uv-straight graph satisfying Conditions (a) and (b) of Lemma 4.5(2) with respect to the original G. If no uv-trail is reported in the previous step, then apply Lemma 4.6 in $\tilde{O}(n^{2\omega})$ time to obtain the data structure D for the winged quadruples of $V(G)$ in G. It remains to show an $O(n^2 \log^2 n)$-time degree-two uv-trailblazer for the uv-straight graph G based on the precomputed D which already spends $O(n^{4.75})$ time. We proceed in two phases. Phase 1 shows that we already have an $O(n^3)$-time degree-two uv-trailblazer for G. Phase 2 then reduces the time to $O(n^2 \log^2 n)$ via Lemma 4.7.

Phase 1 Let B_1 be the $O(n^3)$-time subroutine, taking a pair (a,b) of $V(G)$ as the only argument, that runs the following $O(n^2)$-time procedure for each vertex c of G: Determine from D in $O(n)$ time whether G admits a winged quadruple (a,b,c,d_c) of $V(G)$ for some d_c. If not, then exit. Otherwise, obtain from D in $O(n)$ time a pair (W_1,W_2) of wings for an arbitrary winged (a,b,c,d_c). Since G is uv-straight, it takes $O(n^2)$ time to obtain a monotone uc-path S_c of G containing W_1 and a monotone bv-path T_c of G containing W_2. Obtain in $O(n^2)$ time the subgraph G_c of G induced

doi:10.6342/NTU202203481

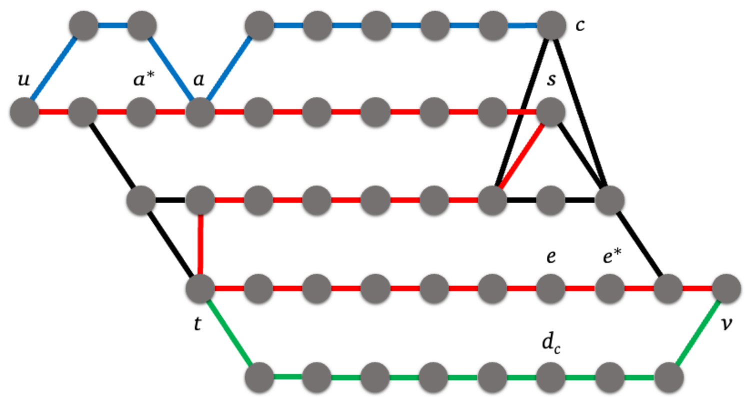


Figure 4.5: An illustration for the proof that B_1 is a uv-trailblazer of degree two. The red path denotes a shortest uv-trail P of the uv-straight graph G. The blue and green paths denote a monotone uc-path S_c and a monotone tv-path T_c of G containing a precomputed pair of wings for (a, t, c, d_c) that need not coincide with P except at a and t.

by

$$(\{x \in V(G) : h(b) \leq h(x) \leq h(c)\} \setminus (N_G[S_c - c] \setminus \{c\})) \cup V(T_c).$$

If the vertices c and b are not connected in G_c, then exit. Otherwise, report the $O(n^2)$-time obtainable concatenation P_c of the uc-path S_c of G and a shortest cv-path of G_c.

By definition of S_c, T_c, and G_c, the uv-path P_c of G reported by $B_1(a, b)$ for any c is induced in G. Since the height of each neighbor of c in G_c is at most $h(c)$, P_c is not monotone. Thus, P_c is a uv-trail of G. Let P be an arbitrary unknown shortest uv-trail of G with twist pair (s, t). Let a (respectively, e) be the vertex of the monotone $P[u, s]$ (respectively, $P[t, v]$) with $h(a) = h(t)$ (respectively, $h(e) = h(s)$). See Figure 4.5 for an illustration. To ensure that B_1 is an $O(n^3)$-time uv-trailblazer of degree 2 for G, the rest of the phase proves that (a, t) is a trail marker for B_1 by showing that the iteration with $c = s$ reports a uv-trail P_s of G.

Let a^* be the neighbor of a in the monotone $P[u, a]$, implying $h(a^*) = h(t) - 1$. Let e^* be the neighbor of e in the monotone $P[e, v]$, implying $h(e^*) = h(s) + 1$. Since $P[a^*, s]$ and $P[t, e^*]$ form

doi:10.6342/NTU202203481

a pair of wings for (a, t, s, e) in G, there is a d_s such that (a, t, s, d_s) is winged in G. Let (W_1, W_2) be the pair of wings for (a, t, s, d_s) in G obtained from D. The monotone us-path S_s of G containing W_1 is a sidetrack for P, since the monotone tv-path T_s of G containing W_2 satisfies Conditions T1 and T2 for S_s. By Lemma 4.3, each vertex x of $P[s, t]$ satisfies $h(t) \leq h(x) \leq h(s)$. By Lemma 4.4, $S_s - s$ and $P[s, t] - s$ are anticomplete in G, implying that $P[s, t]$ is a path of G_s. Since s and t are connected in G_s, the subroutine call $B_1(a, t)$ outputs a uv-trail P_s of G in the iteration with $c = s$. Hence, (a, t) is indeed a trail marker of B. One can verify that P_s is also a shortest uv-trail of the uv-straight G, although d_s need not be e. Thus, we have an $O(n^5)$-time algorithm on an n-vertex general (respectively, uv-straight) graph G that either reports a general (respectively, shortest) uv-trail of G or ensures that G is uv-trailless.

Phase 2 Since many prefixes of a long sidetrack for a shortest uv-trail P of G remain sidetracks for P, an edge can be deleted and then inserted back $\Omega(n)$ times in Phase 1. Phase 2 avoids the redundancy by processing the sidetracks in the decreasing order of their lengths. Let B_2 be the following subroutine that takes a pair (a, b) of $V(G)$ as the only argument. Obtain in overall $O(n^2)$ time from D each set C_i with $0 \leq i \leq h(v)$ that consists of the vertices c of G with $h(c) = i$ such that G admits a winged quadruple (a, b, c, d_c) for some vertex d_c. Let C be the union of all C_i with $0 \leq i \leq h(v)$. Obtain in overall $O(n^2)$ time from D for each vertex $c \in C$ (i) a monotone uc-path S_c of G containing a and (ii) a monotone bv-path T_c with

$$d_{G[S_c \cup T_c]}(u, v) > \|S_c\| + \|T_c\|.$$

Obtain the subgraph H of G induced by the vertices with heights at least $h(a)$ in $O(n^2 \log^2 n)$ time by the dynamic data structure of Lemma 4.7. Iteratively perform the following steps in the decreasing order of the indices i with $h(a) \leq i < h(v)$:

1. Delete from H the incident edges of $N_G[S_c - c] \setminus \{c\}$ in G for all $c \in C_i$.

doi:10.6342/NTU202203481

2. Insert to H the incident edges of C_i in G.

3. Delete from H all edges xy of G with $h(x) = i$ and $h(y) = i + 1$.

4. If b is not connected to any $c \in C_i$ in H, then proceed to the next iteration. Otherwise, let c be an arbitrary vertex of C_i that is connected to b in H. Exit the loop and report the $O(n^2)$-time obtainable concatenation P_c of S_c and a shortest cv-path of $G[H \cup T_c]$.

Since $S_c - c$ and $T_c - b$ are anticomplete in G and the height of each neighbor of c in H is at most $h(c)$, any arbitrary reported uv-path P_c of G is a uv-trail of G.

Throughout all iterations, the incident edges of each vertex of G is deleted $O(1)$ times by Step 1, each edge of G is updated $O(1)$ times by Steps 2 and 3, and each vertex $c \in C$ is queried $O(1)$ times by Step 4. Thus, each subroutine call $B_2(a, b)$ runs in $O(n^2 \log^2 n)$ time.

Let P be an arbitrary shortest uv-trail of G with twist pair (s, t). As in Phase 1, let a (respectively, e) be the vertex of the monotone $P[u, s]$ (respectively, $P[t, v]$) with $h(a) = h(t)$ (respectively, $h(e) = h(s)$). The rest of the phase proves that (a, t) is a trail marker for B_2 by showing that an iteration with $i \geq h(s)$ in the loop of the subroutine call $B_2(a, t)$ reports a uv-trail P_c of G. See Figures 4.6 and 4.7 for an illustration.

If an iteration of $B_2(a, t)$ with $i \geq h(s) + 1$ reports a uv-trail of G (that need not be shortest), then we are done. Otherwise, we show that the iteration with $i = h(s)$ has to report a uv-trail of G. For each $c \in C$ with $h(c) \geq i$, let s_c be the unknown vertex of S_c with $h(s_c) = i$. $S_c[u, s_c]$ remains a sidetrack for P, since T_c still satisfies Conditions T1 and T2 for $S_c[u, s_c]$. Thus, $s_c \in C_i$. By Lemma 4.4, $S_c[u, s_c] - s_c$ and $P[s, t] - s$ are anticomplete in G even if $S_c[u, s_c]$ need not be S_{s_c}. As a result, $P[s, t] - s$ is a path of the H at the completion of Step 1 in the i-th iteration. $s \in C_i$ and Lemma 4.3, $P[s, t]$ is a path of the graph H at the completion of Step 3 in the i-th iteration. Therefore, s is a $c \in C_i$ that is connected to t in H. Step 4 in the i-th iteration has to output a (shortest) uv-trail P_c of G for some $c \in C_i$ that need not be s. Thus, we have an $O(n^{4.75})$-time algorithm that either obtains a uv-trail of G or ensures that G is uv-trailless. A reported uv-trail

doi:10.6342/NTU202203481

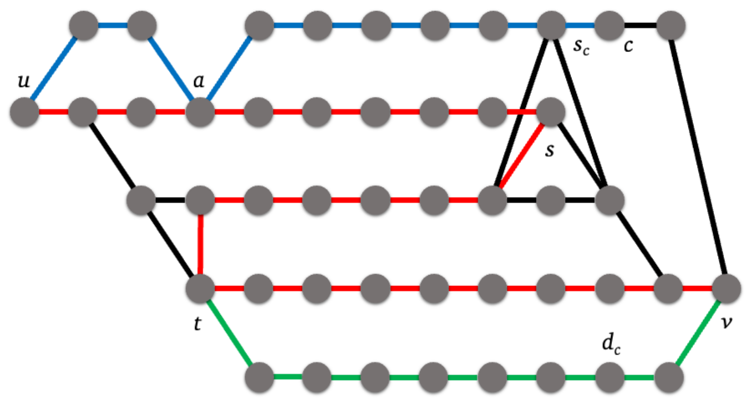


Figure 4.6: An illustration for the proof that B_2 is a uv-trailblazer of degree two. The red path denotes a shortest uv-trail P of the uv-straight graph G. The blue and green paths denote a monotone uc-path S_c and a monotone tv-path T_c of G containing a precomputed pair of wings for (a, t, c, d_c) that need not coincide with P except at a and t. $S_c[u, s_c]$ remains a sidetrack for P.

of G by this $O(n^{4.75})$-time algorithm need not be a shortest uv-trail of G, since we cannot afford to spend $O(n^2)$ time, as in Phase 1, for each $c \in C$ that is connected to t in the H at the $h(c)$-th iteration to obtain a shortest cv-path of $G[H \cup T_c]$.

doi:10.6342/NTU202203481

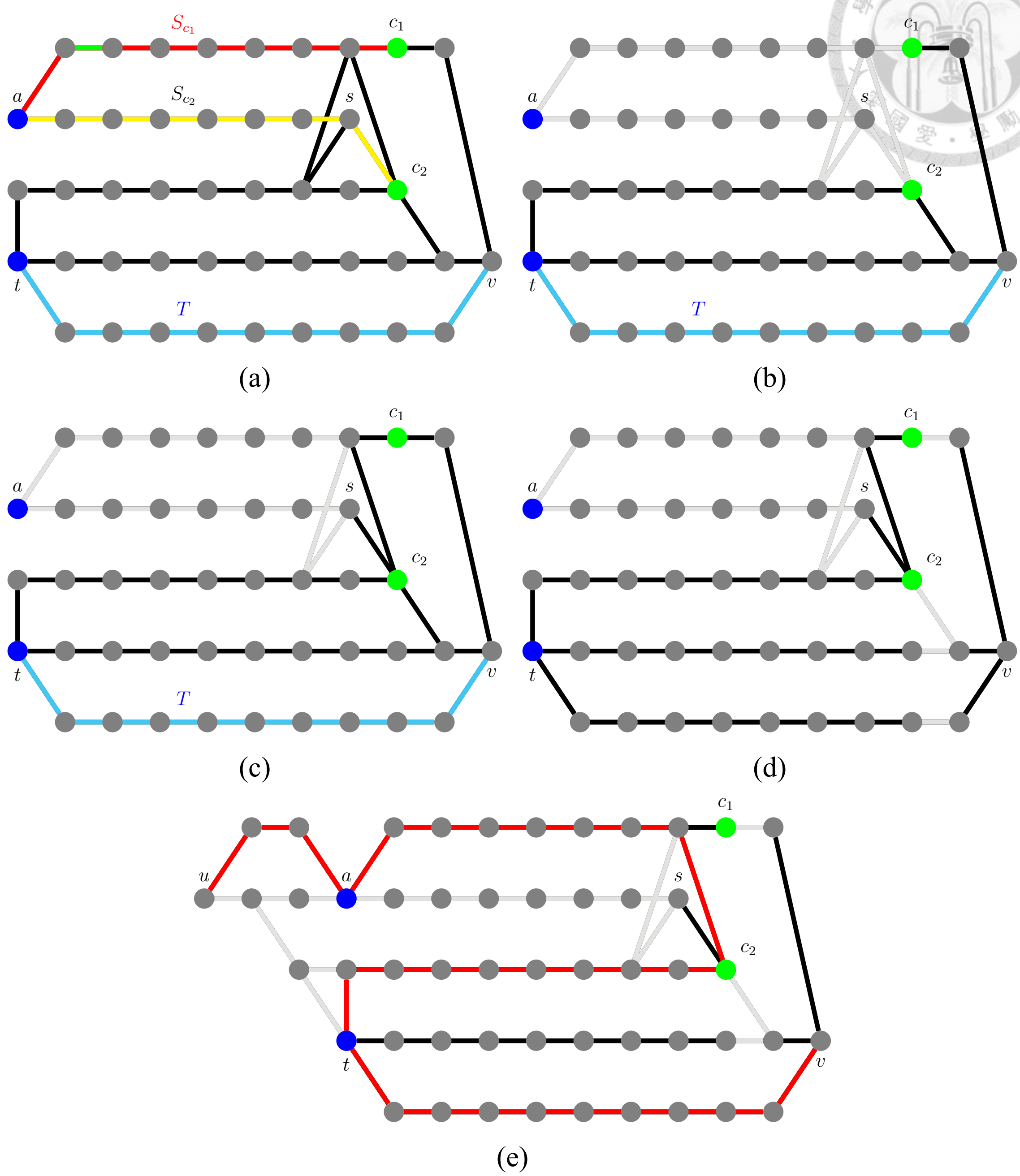


Figure 4.7: An illustration for running $B_2(a, t)$ on Figure 4.6. (a) The initial H. (b) The H after Step 1. (c) The H after Step 2. (d) The H after Step 3 in which c_2 and t are connected. (e) The H after Step 4. The red path is a non-shortest uv-trail of G reported by $B_2(a, t)$.

doi:10.6342/NTU202203481

Chapter 5

Conclusion

Algorithms for induced subgraphs are important and challenging. We give improved algorithms for (1) recognizing perfect graphs via detecting odd holes, (2) finding a shortest odd hole, and (3) finding a trail between two given vertices. It is of interest to further reduce the required time of these three problems.

We achieve the improvement of algorithm for (1) by showing that guessing at most 5 vertices suffices for the bottleneck subroutine of the odd-hole detection algorithm of [65] to pin down an odd hole of G.

As for the algorithm for (2), we achieve the improvement by a new observation described by Claim 1 in the proof of Lemma 3.1 that guessing 11 vertices suffices for the bottleneck subroutine of the shortest-odd-hole detection algorithm of [23] to pin down a shortest odd hole of G. It is of interest to know whether some vertices of a marker of T_i for a tripod T of G can be removed from the list of guessed vertices so that a further improved algorithm might be feasible.

The key to our improved algorithm for (3) is the observation regarding an arbitrary shortest uv-trail of a uv-straight graph G described by Lemma 4.4. The inequality of Lemma 4.4 is stronger than

doi:10.6342/NTU202203481

the condition that $S - c$ and $P[s,t] - s$ are anticomplete in G. As a matter of fact, the latter suffices for our uv-trailblazers in §4.2 and §4.3. Thus, a further improved uv-trailblazer might be possible if the wings for a winged quadruple can be obtained more efficiently. As mentioned in Phase 1 of §4.3, a shortest uv-trail, if any, of a uv-straight G can be obtained by our B_1-based trailblazing algorithm in $O(n^5)$ time. Detecting a uv-trail with length at least $2d_G(u,v)$ is NP-complete [7, Theorem 1.6]. It would be of interest to see if a shortest uv-trail or a uv-trail having length at least $d_G(u,v)+k$ for a positive $k = O(1)$ in a general G can be obtained in polynomial time. It is also of interest to see whether the one-to-all (respectively, all-pairs) version of the problem can be solved in time much lower than $O(n^{5.75})$ (respectively, $O(n^{6.75})$).

doi:10.6342/NTU202203481

References

[1] T. Abrishami, M. Chudnovsky, M. Pilipczuk, P. Rzazewski, and P. D. Seymour. Induced subgraphs of bounded treewidth and the container method. In D. Marx, editor, *Proceedings of the 32nd ACM-SIAM Symposium on Discrete Algorithms*, pages 1948–1964, 2021. doi:10.1137/1.9781611976465.116.

[2] J. Alman and V. Vassilevska Williams. A refined laser method and faster matrix multiplication. In D. Marx, editor, *Proceedings of the 32nd ACM-SIAM Symposium on Discrete Algorithms*, pages 522–539, 2021. doi:10.1137/1.9781611976465.32.

[3] N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

[4] C. Berge. Les problèmes de coloration en théorie des graphes. *Publications de l'Institut de statistique de l'Université de Paris*, 9:123–160, 1960.

[5] C. Berge. Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung). *Wissenschaftliche Zeitschrift, Martin Luther Universität Halle-Wittenberg, Mathematisch-Naturwissenschaftliche Reihe*, 10:114–115, 1961.

[6] C. Berge. *Graphs*. North-Holland, Amsterdam, New York, 1985.

[7] E. Berger, P. D. Seymour, and S. Spirkl. Finding an induced path that is not a shortest path. *Discrete Mathematics*, 344(7):112398.1–112398.6, 2021. doi:10.1016/j.disc.2021.112398.

[8] D. Bienstock. On the complexity of testing for odd holes and induced odd paths. *Discrete Mathematics*, 90(1):85–92, 1991. doi:10.1016/0012-365X(91)90098-M, see [9] for corrigendum.

[9] D. Bienstock. Corrigendum to: D. Bienstock, "On the complexity of testing for odd holes and induced odd paths" Discrete Mathematics 90 (1991) 85–92. *Discrete Mathematics*, 102(1):109, 1992. doi:10.1016/0012-365X(92)90357-L.

[10] N. Biggs, E. K. Lloyd, and R. J. Wilson. *Graph Theory, 1736-1936*. Oxford University Press, 1986.

[11] A. Björklund, T. Husfeldt, and P. Kaski. The shortest even cycle problem is tractable. In S. Leonardi and A. Gupta, editors, *Proceedings of the 54th Annual Symposium on Theory of Computing*, pages 117–130, 2022. doi:10.1145/3519935.3520030.

[12] H.-C. Chang and H.-I. Lu. Computing the girth of a planar graph in linear time. *SIAM Journal on Computing*, 42(3):1077–1094, 2013. doi:10.1137/110832033.

[13] H.-C. Chang and H.-I. Lu. A faster algorithm to recognize even-hole-free graphs. *Journal of Combinatorial Theory, Series B*, 113:141–161, 2015. doi:10.1016/j.jctb.2015.02.001.

[14] Y. Chen and J. Flum. On parameterized path and chordless path problems. In *Proceedings of the 22nd Annual IEEE Conference on Computational Complexity*, pages 250–263, 2007. doi:10.1109/CCC.2007.21.

[15] H.-T. Cheong and H.-I. Lu. Finding a shortest even hole in polynomial time. *Journal of Graph Theory*, 99(3):425–434, 2022. doi:10.1002/jgt.22748.

[16] Y.-C. Chiu and H.-I. Lu. Blazing a trail via matrix multiplications: A faster algorithm for non-shortest induced paths. In P. Berenbrink and B. Monmege, editors, *Proceedings of the*

39th International Symposium on Theoretical Aspects of Computer Science, LIPIcs 219, pages 23:1–23:16, 2022. doi:10.4230/LIPIcs.STACS.2022.23.

[17] M. Chudnovsky, G. Cornuéjols, X. Liu, P. D. Seymour, and K. Vušković. Recognizing Berge graphs. *Combinatorica*, 25(2):143–186, 2005. doi:10.1007/s00493-005-0012-8.

[18] M. Chudnovsky, K.-i. Kawarabayashi, and P. Seymour. Detecting even holes. *Journal of Graph Theory*, 48(2):85–111, 2005. doi:10.1002/jgt.20040.

[19] M. Chudnovsky, M. Pilipczuk, M. Pilipczuk, and S. Thomassé. On the maximum weight independent set problem in graphs without induced cycles of length at least five. *SIAM Journal on Discrete Mathematics*, 34(2):1472–1483, 2020. doi:10.1137/19M1249473.

[20] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164(1):51–229, 2006. doi:10.4007/annals.2006.164.51.

[21] M. Chudnovsky, N. Robertson, P. D. Seymour, and R. Thomas. Progress on perfect graphs. *Mathematical Programming*, 97(1-2):405–422, 2003. doi:10.1007/s10107-003-0449-8.

[22] M. Chudnovsky, A. Scott, and P. Seymour. Detecting a long odd hole. *Combinatorica*, 41(1):1–30, 2021. doi:10.1007/s00493-020-4301-z.

[23] M. Chudnovsky, A. Scott, and P. Seymour. Finding a shortest odd hole. *ACM Transactions on Algorithms*, 17(2):13.1–13.21, 2021. doi:10.1145/3447869.

[24] M. Chudnovsky, A. Scott, P. Seymour, and S. Spirkl. Detecting an odd hole. *Journal of the ACM*, 67(1):5.1–5.12, 2020. doi:10.1145/3375720.

[25] M. Chudnovsky and P. Seymour. The three-in-a-tree problem. *Combinatorica*, 30(4):387–417, 2010. doi:10.1007/s00493-010-2334-4.

[26] M. Chudnovsky and P. D. Seymour. Even pairs in Berge graphs. *Journal of Combinatorial Theory, Series B*, 99(2):370–377, 2009. doi:10.1016/j.jctb.2008.08.002.

[27] M. Chudnovsky and V. Sivaraman. Odd holes in bull-free graphs. *SIAM Journal on Discrete Mathematics*, 32(2):951–955, 2018. doi:10.1137/17M1131301.

[28] V. Chvátal and N. Sbihi. Bull-free berge graphs are perfect. *Graphs and Combinatorics*, 3(1):127–139, 1987. doi:10.1007/BF01788536.

[29] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Finding an even hole in a graph. In *Proceedings of the 38th Symposium on Foundations of Computer Science*, pages 480–485, 1997. doi:10.1109/SFCS.1997.646136.

[30] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Even and odd holes in cap-free graphs. *Journal of Graph Theory*, 30(4):289–308, 1999. doi:10.1002/(SICI)1097-0118(199904)30:4<289::AID-JGT4>3.0.CO;2-3.

[31] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Even-hole-free graphs Part I: Decomposition theorem. *Journal of Graph Theory*, 39(1):6–49, 2002. doi:10.1002/jgt.10006.

[32] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Even-hole-free graphs Part II: Recognition algorithm. *Journal of Graph Theory*, 40(4):238–266, 2002. doi:10.1002/jgt.10045.

[33] M. Conforti, G. Cornuéjols, X. Liu, K. Vušković, and G. Zambelli. Odd hole recognition in graphs of bounded clique size. *SIAM Journal on Discrete Mathematics*, 20(1):42–48, 2006. doi:10.1137/S089548010444540X.

[34] M. Conforti, G. Cornuéjols, and M. R. Rao. Decomposition of balanced matrices. *Journal of Combinatorial Theory, Series B*, 77(2):292–406, 1999. doi:10.1006/jctb.1999.1932.

[35] M. Conforti, G. Cornuéjols, and K. Vušković. Square-free perfect graphs. *Journal of Combinatorial Theory, Series B*, 90(2):257–307, 2004. doi:10.1016/j.jctb.2003.08.003.

[36] L. Cook, J. Horsfield, M. Preissmann, C. Robin, P. Seymour, N. L. D. Sintiari, N. Trotignon, and K. Vušković. Graphs with all holes the same length. *arXiv*, 2021. doi:10.48550/arxiv.2110.09970.

[37] L. Cook and P. D. Seymour. Detecting a long even hole. *European Journal of Combinatorics*, 104:103537, 2022. doi:10.1016/j.ejc.2022.103537.

[38] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990. doi:10.1016/S0747-7171(08)80013-2.

[39] G. Cornuéjols, X. Liu, and K. Vušković. A polynomial algorithm for recognizing perfect graphs. In *Proceedings of the 44th Symposium on Foundations of Computer Science*, pages 20–27, 2003. doi:10.1109/SFCS.2003.1238177.

[40] M. V. G. da Silva and K. Vušković. Decomposition of even-hole-free graphs with star cutsets and 2-joins. *Journal of Combinatorial Theory, Series B*, 103(1):144–183, 2013. doi:10.1016/j.jctb.2012.10.001.

[41] S. Dahlgaard, M. B. T. Knudsen, and M. Stöckel. Finding even cycles faster via capped k-walks. In H. Hatami, P. McKenzie, and V. King, editors, *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, pages 112–120. ACM, 2017. doi:10.1145/3055399.3055459.

[42] M. Dalirrooyfard and V. Vassilevska Williams. Induced cycles and paths are harder than you think. In *Proceedings of the 63rd Annual Symposium on Foundations of Computer Science*, 2022, to appear. doi:10.48550/arXiv.2209.01873.

[43] M. Dalirrooyfard, T. D. Vuong, and V. Vassilevska Williams. Graph pattern detection: hardness for all induced patterns and faster non-induced cycles. In *Proceedings of the 51st Symposium on Theory of Computing*, pages 1167–1178, 2019. doi:10.1145/3313276.3316329.

[44] D. Eppstein. Finding the k shortest paths. *SIAM Journal on Computing*, 28(2):652–673, 1998. doi:10.1137/S0097539795290477.

[45] H. Everett, C. M. H. de Figueiredo, C. L. Sales, F. Maffray, O. Porto, and B. A. Reed. Path parity and perfection. *Discrete Mathematics*, 165-166:233–252, 1997. doi:10.1016/S0012-365X(96)00174-4.

[46] M. R. Fellows, J. Kratochvíl, M. Middendorf, and F. Pfeiffer. The complexity of induced minors and related problems. *Algorithmica*, 13(3):266–282, 1995. doi:10.1007/BF01190507.

[47] J. Fonlupt and J. Uhry. Transformations which preserve perfectness and H-perfectness of graphs. In A. Bachem, M. Grötschel, and B. Korte, editors, *Bonn Workshop on Combinatorial Optimization*, volume 66 of *North-Holland Mathematics Studies*, pages 83–95. North-Holland, 1982. doi:10.1016/S0304-0208(08)72445-9.

[48] Z. Galil and O. Margalit. Witnesses for boolean matrix multiplication and for transitive closure. *Journal of Complexity*, 9(2):201–221, 1993. doi:10.1006/jcom.1993.1014.

[49] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[50] S. Gaspers, S. Huang, and D. Paulusma. Colouring square-free graphs without long induced paths. In R. Niedermeier and B. Vallée, editors, *Proceedings of the 35th Symposium on Theoretical Aspects of Computer Science*, LIPIcs 96, pages 35.1–35.15, 2018. doi:10.4230/LIPIcs.STACS.2018.35.

[51] E. D. Giacomo, G. Liotta, and T. Mchedlidze. Lower and upper bounds for long induced paths in 3-connected planar graphs. *Theoretical Computer Science*, 636:47–55, 2016. doi:10.1016/j.tcs.2016.04.034.

[52] P. A. Golovach, D. Paulusma, and E. J. van Leeuwen. Induced disjoint paths in AT-free

doi:10.6342/NTU202203481

graphs. In F. V. Fomin and P. Kaski, editors, *Proceedings of the 13th Scandinavian Symposium and Workshops on Algorithm Theory*, Lecture Notes in Computer Science 7357, pages 153–164, 2012. doi:10.1007/978-3-642-31155-0_14.

[53] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[54] R. Haas and M. Hoffmann. Chordless paths through three vertices. *Theoretical Computer Science*, 351(3):360–371, 2006. doi:10.1016/j.tcs.2005.10.021.

[55] C. T. Hoàng, M. Kaminski, J. Sawada, and R. Sritharan. Finding and listing induced paths and cycles. *Discrete Applied Mathematics*, 161(4-5):633–641, 2013. doi:10.1016/j.dam.2012.01.024.

[56] J. Holm, K. de Lichtenberg, and M. Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *Journal of the ACM*, 48(4):723−760, July 2001. doi:10.1145/502090.502095.

[57] W.-L. Hsu. Recognizing planar perfect graphs. *Journal of the ACM*, 34(2):255–288, 1987. doi:10.1145/23005.31330.

[58] A. Itai and M. Rodeh. Finding a minimum circuit in a graph. *SIAM Journal on Computing*, 7(4):413–423, 1978. doi:10.1137/0207033.

[59] L. Jaffke, O.-j. Kwon, and J. A. Telle. Mim-width I. induced path problems. *Discrete Applied Mathematics*, 278:153–168, 2020. doi:10.1016/j.dam.2019.06.026.

[60] D. S. Johnson. The NP-completeness column. *ACM Transactions on Algorithms*, 1(1):160–176, 2005. doi:10.1145/1077464.1077476.

[61] M. Kaminski and N. Nishimura. Finding an induced path of given parity in planar graphs in polynomial time. In Y. Rabani, editor, *Proceedings of the 23rd Annual ACM-SIAM Sympo-*

sium on Discrete Algorithms, pages 656–670, 2012. doi:10.1137/1.9781611973099.55.

[62] K. Kawarabayashi, Y. Kobayashi, and B. A. Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 102(2):424–435, 2012. doi:10.1016/j.jctb.2011.07.004.

[63] L. G. Khachiyan. A polynomial algorithm in linear programming. In *Doklady Akademii Nauk*, volume 244, pages 1093–1096. Russian Academy of Sciences, 1979.

[64] M. Kriesell. Induced paths in 5-connected graphs. *Journal of Graph Theory*, 36(1):52–58, 2001. doi:10.1002/1097-0118(200101)36:1<52::AID-JGT5>3.0.CO;2-N.

[65] K.-Y. Lai, H.-I. Lu, and M. Thorup. Three-in-a-tree in near linear time. In *Proccedings of the 52nd Annual ACM Symposium on Theory of Computing*, pages 1279–1292, 2020. doi:10.1145/3357713.3384235.

[66] F. Le Gall. Powers of tensors and fast matrix multiplication. In K. Nabeshima, K. Nagasaka, F. Winkler, and Á. Szántó, editors, *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 296–303, 2014. doi:10.1145/2608628.2608664.

[67] W. Liu and N. Trotignon. The k-in-a-tree problem for graphs of girth at least k. *Discrete Applied Mathematics*, 158(15):1644–1649, 2010. doi:10.1016/j.dam.2010.06.005.

[68] L. Lovász. A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B*, 13(2):95–98, 1972. doi:10.1016/0095-8956(72)90045-7.

[69] L. Lovász. Graph minor theory. *American Mathematical Society. Bulletin. New Series*, 43(1):75–86, 2006. doi:10.1090/S0273-0979-05-01088-8.

[70] W. McCuaig. Pólya's permanent problem. *Electronic Journal of Combinatorics*, 11(1), 2004. doi:10.37236/1832.

[71] H. Meyniel. A new property of critical imperfect graphs and some consequences. *European Journal of Combinatorics*, 8(3):313–316, 1987. doi:10.1016/S0195-6698(87)80037-9.

[72] O. Porto. Even induced cycles in planar graphs. In *Proceedings of the 1st Latin American Symposium on Theoretical Informatics*, pages 417–429, 1992. doi:10.1007/BFb0023845.

[73] M. Radovanovic, N. Trotignon, and K. Vušković. The (theta, wheel)-free graphs part IV: induced paths and cycles. *Journal of Combinatorial Theory, Series B*, 146:495–531, 2021. doi:10.1016/j.jctb.2020.06.002.

[74] B. A. Reed. A semi-strong perfect graph theorem. *Journal of Combinatorial Theory, Series B*, 43(2):223–240, 1987. doi:10.1016/0095-8956(87)90022-0.

[75] N. Robertson and P. D. Seymour. Graph minors. I. Excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983. doi:10.1016/0095-8956(83)90079-5.

[76] N. Robertson and P. D. Seymour. Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory, Series B*, 36(1):49–64, 1984. doi:10.1016/0095-8956(84)90013-3.

[77] N. Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986. doi:10.1016/0196-6774(86)90023-4.

[78] N. Robertson and P. D. Seymour. Graph minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, 1986. doi:10.1016/0095-8956(86)90030-4.

[79] N. Robertson and P. D. Seymour. Graph minors. VI. Disjoint paths across a disc. *Journal of Combinatorial Theory, Series B*, 41(1):115–138, 1986. doi:10.1016/0095-8956(86)90031-6.

[80] N. Robertson and P. D. Seymour. Graph minors. VII. Disjoint paths on a surface. *Journal of Combinatorial Theory, Series B*, 45(2):212–254, 1988. doi:10.1016/0095-8956(88)90070-6.

[81] N. Robertson and P. D. Seymour. Graph minors. IV. Tree-width and well-quasi-ordering. *Journal of Combinatorial Theory, Series B*, 48(2):227–254, 1990. doi:10.1016/0095-

8956(90)90120-O.

[82] N. Robertson and P. D. Seymour. Graph minors. IX. Disjoint crossed paths. *Journal of Combinatorial Theory, Series B*, 49(1):40–77, 1990. doi:10.1016/0095-8956(90)90063-6.

[83] N. Robertson and P. D. Seymour. Graph minors. VIII. A Kuratowski theorem for general surfaces. *Journal of Combinatorial Theory, Series B*, 48(2):255–288, 1990. doi:10.1016/0095-8956(90)90121-F.

[84] N. Robertson and P. D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991. doi:10.1016/0095-8956(91)90061-N.

[85] N. Robertson and P. D. Seymour. Graph minors. XI. Circuits on a surface. *Journal of Combinatorial Theory, Series B*, 60(1):72–106, 1994. doi:10.1006/jctb.1994.1007.

[86] N. Robertson and P. D. Seymour. Graph minors. XII. Distance on a surface. *Journal of Combinatorial Theory, Series B*, 64(2):240–272, 1995. doi:10.1006/jctb.1995.1034.

[87] N. Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995. doi:10.1006/jctb.1995.1006.

[88] N. Robertson and P. D. Seymour. Graph minors. XIV. Extending an embedding. *Journal of Combinatorial Theory, Series B*, 65(1):23–50, 1995. doi:10.1006/jctb.1995.1042.

[89] N. Robertson and P. D. Seymour. Graph minors: XV. Giant steps. *Journal of Combinatorial Theory, Series B*, 68(1):112–148, 1996. doi:10.1006/jctb.1996.0059.

[90] N. Robertson and P. D. Seymour. Graph minors: XVII. Taming a vortex. *Journal of Combinatorial Theory, Series B*, 77(1):162–210, 1999. doi:10.1006/jctb.1999.1919.

[91] N. Robertson and P. D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43–76, 2003. doi:10.1016/S0095-8956(03)00042-

X.

[92] N. Robertson and P. D. Seymour. Graph minors. XVIII. Tree-decompositions and well-quasi-ordering. *Journal of Combinatorial Theory, Series B*, 89(1):77–108, 2003. doi:10.1016/S0095-8956(03)00067-4.

[93] N. Robertson and P. D. Seymour. Graph minors. XIX. Well-quasi-ordering on a surface. *Journal of Combinatorial Theory, Series B*, 90(2):325–385, 2004. doi:10.1016/j.jctb.2003.08.005.

[94] N. Robertson and P. D. Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004. doi:10.1016/j.jctb.2004.08.001.

[95] N. Robertson and P. D. Seymour. Graph minors. XXI. Graphs with unique linkages. *Journal of Combinatorial Theory, Series B*, 99(3):583–616, 2009. doi:10.1016/j.jctb.2008.08.003.

[96] N. Robertson and P. D. Seymour. Graph minors XXIII. Nash-Williams' immersion conjecture. *Journal of Combinatorial Theory, Series B*, 100(2):181–205, 2010. doi:10.1016/j.jctb.2009.07.003.

[97] N. Robertson and P. D. Seymour. Graph minors. XXII. Irrelevant vertices in linkage problems. *Journal of Combinatorial Theory, Series B*, 102(2):530–563, 2012. doi:10.1016/j.jctb.2007.12.007.

[98] N. Robertson, P. D. Seymour, and R. Thomas. Permanents, Pfaffian orientations, and even directed circuits. *Annals of Mathematics*, 150(3):929–975, 1999. doi:10.2307/121059.

[99] D. Rose, R. Tarjan, and G. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976. doi:10.1137/0205021.

[100] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on*

Computing, 13(3):566–579, 1984. doi:10.1137/0213035, see [101] for addendum.

[101] R. E. Tarjan and M. Yannakakis. Addendum: Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 14(1):254–255, 1985. doi:10.1137/0214020.

[102] N. Trotignon. *Perfect graphs*, page 137–160. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2015.

[103] V. Vassilevska Williams. Multiplying matrices faster than Coppersmith–Winograd. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 887–898, 2012. doi:10.1145/2213977.2214056.

[104] V. V. Williams and R. R. Williams. Subcubic equivalences between path, matrix, and triangle problems. *Journal of the ACM*, 65(5):27:1–27:38, 2018. doi:10.1145/3186893.

[105] R. Yuster and U. Zwick. Finding even cycles even faster. *SIAM Journal on Discrete Mathematics*, 10(2):209–222, 1997. doi:10.1137/S0895480194274133.