# A Shorter Proof of the Graph Minor Algorithm – The Unique Linkage Theorem –*

[Extended Abstract] †

Ken-ichi Kawarabayashi‡
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo, Japan
k_keniti@nii.ac.jp

Paul Wollan
Department of Computer Science
University of Rome, *La Sapienza*
Rome, Italy
wollan@di.uniroma1.it

## ABSTRACT

At the core of the seminal Graph Minor Theory of Robertson and Seymour is a powerful theorem which describes the structure of graphs excluding a fixed minor. This result is used to prove Wagner's conjecture and provide a polynomial time algorithm for the disjoint paths problem when the number of the terminals is fixed (i.e, the Graph Minor Algorithm). However, both results require the full power of the Graph Minor Theory, i.e, the structure theorem.

In this paper, we show that this is not true in the latter case. Namely, we provide a new and much simpler proof of the correctness of the Graph Minor Algorithm. Specifically, we prove the "Unique Linkage Theorem" without using Graph Minors structure theorem. The new argument, in addition to being simpler, is much shorter, cutting the proof by at least 200 pages. We also give a new full proof of correctness of an algorithm for the well-known edge-disjoint paths problem when the number of the terminals is fixed, which is at most 25 pages long.

## Categories and Subject Descriptors

G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms, Path and circuit problems*

## General Terms

Algorithms, Theory

*This work was partially supported by MOU grant at National Institute of Informatics.

†A full version of this paper is available at `http://reseach.nii.ac.jp/~k_keniti/uniquelink.pdf`

‡Research partly supported by Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research, by C and C Foundation, by Kayamori Foundation and by Inoue Research Award for Young Scientists.


*STOC'10,* June 5–8, 2010, Cambridge, Massachusetts, USA.


## Keywords

Disjoint paths problem, Graph Minors, linkages

## 1. INTRODUCTION

### 1.1 Graph Minors Algorithm

One of the deepest and most important bodies of work in graph theory is the Graph Minor Theory developed by Robertson and Seymour. At the heart of this theory is a theorem [33, Theorem 1.3] describing the structure of all graphs excluding a fixed graph as a minor. At a high level, the theorem says that every such graph can be decomposed into a collection of graphs each of which can "almost" be embedded into a bounded-genus surface, combined in a tree structure. Much of the Graph Minors series of articles is devoted to the proof of this structure theorem.

The main algorithmic result of the Graph Minor Theory is a polynomial-time algorithm for testing the existence of a fixed minor [31] which, combined with the proof of Wagner's Conjecture, implies the existence of a polynomial-time algorithm for deciding any minor-closed graph property. The existence of such a polynomial time algorithm has in turn been used to show the existence of polynomial-time algorithms for several graph problems, some of which were not previously known to be decidable [10]. It also leads to the framework of parameterized complexity developed by Downey and Fellows [8], which is perhaps one of the most active areas in the study of algorithms.

This algorithm is one of the most important polynomial time algorithms in theoretical computer science. The algorithm is relatively easy to describe. However, the proof of correctness of the algorithm (that is, the proof that the algorithm does in fact correctly determine the presence of a fixed graph as a minor) uses the full power of the Graph Minor Theory. More precisely, we can immediately reduce the problem to the case when the input graph has no large clique minor. However, the analysis of this case requires the development of the structure theorem. Our goal is to provide a new proof for the correctness of this algorithm that avoids many of the difficulties and technicalities in the original proof of Robertson and Seymour, and, specifically, avoids the use of the structure theorem.

The main purpose of this paper is to show the correctness of the Graph Minor Algorithm without using the structure

theorem. This leads to a dramatically shorter and more simple proof of the correctness for the algorithm.

Much of the proof of the correctness of Graph Minor Algorithm in fact focuses on developing an algorithm for the disjoint paths problem. It will be more convenient for us, as well, to focus on the disjoint paths problem. We discuss this further in the next subsection.

## 1.2 The Graph Minors Algorithm vs. the k-disjoint paths problem

In the edge- (vertex-) disjoint paths problem, we are given a graph $G$ and a set of $k$ pairs of vertices, called *terminals*, in $G$, and we have to decide whether or not $G$ has $k$ edge- (vertex-) disjoint paths connecting given pairs of terminals. The problem of testing whether a given graph contains a fixed graph $H$ as a minor can be trivially reduced to a bounded number of vertex disjoint path problems. Thus, a polynomial time algorithm for the $k$ disjoint paths problem yields a polynomial time algorithm for minor testing, albeit with a worse runtime than that of the Graph Minors Algorithm. Moreover, the arguments for minor testing and the disjoint paths problem are analogous, although somewhat simpler to explain in the case of the disjoint paths problem. Finally, the $k$ disjoint paths problem is also a classic problem the theory of algorithms, widely studied in its own right. For all these reasons, for the remainder of the article we will restrict our attention to the $k$ disjoint paths problem. We will return only briefly to Graph Minors Algorithm to show how our results yield a short argument for the correctness of the Graph Minors Algorithm.

## 1.3 Background on the disjoint paths problem

The $k$ disjoint paths problem, both in it's vertex and edge disjoint versions, is a central problem in algorithmic graph theory and combinatorial optimization. See the surveys [11, 36]. It has attracted attention in the contexts of transportation networks, VLSI layout and virtual circuit routing in high-speed networks or on the internet. A basic technical problem here is to interconnect certain prescribed "channels" on a chip such that wires belonging to different pins do not touch each other. In this simplest form, the problem mathematically amounts to finding disjoint trees in a graph or disjoint paths in a graph, each connecting a given set of vertices.

We now quickly look at previous results on the $k$ disjoint paths problem. If $k$ is as a part of the input of the problem, then this is one of Karp's original NP-complete problems [13], and it remains NP-complete even if $G$ is restricted to be planar (Lynch [22]). The seminal work of Robertson and Seymour says that there exists a polynomial time algorithm (the actual runtime of the algorithm is $O(n^3)$. The time complexity is improved to $O(n^2)$ in [18]) for the disjoint paths problem when the number of terminals, $k$, is fixed. In the next subsection, we give an outline of this algorithm.

## 1.4 Robertson-Seymour Algorithm

In this subsection, we sketch Robertson and Seymour's algorithm for the $k$ disjoint paths problem (see also [27]). At a high level, Robertson-Seymour's algorithm is based on the following two cases: either a given graph $G$ has bounded tree-width (bounded by some function of $k$) or else it has large tree-width.

**Case 1.** Tree-width of $G$ is bounded.

In this case, one can apply a dynamic programming argument to a tree-decomposition of bounded tree-width, see [1, 2, 31].

**Case 2.** Tree-width of $G$ is large.

This second case again breaks into two cases depending on whether $G$ has a large clique minor or not.

**Case 2.1.** $G$ has a large clique minor.

If there exist disjoint paths from the terminals to this clique minor, then we can use this clique minor to link up the terminals in any desired way. Otherwise, there is a small cut set such that the large clique minor is cut off from the terminals by this cut set. In this case, we can prove that there is a vertex $v$ in the clique minor which is *irrelevant*, i.e., the $k$ disjoint paths problem is feasible in $G$ if and only if it is also feasible in $G - v$. We then recursively apply the algorithm to $G - v$.

**Case 2.2.** $G$ does not have a huge clique minor.

In this case, one can prove that, after deleting a bounded number of vertices, there is a huge subgraph which is essentially planar. Moreover, this huge planar subgraph itself has very large tree width. This makes it possible to prove that the middle vertex $v$ of this wall is irrelevant. Again, we recursively apply the algorithm to $G - v$.

The analysis of Cases 1 and 2.1 is relatively easy. It is the analysis of Case 2.2 that gives rise to the majority of the work. The analysis of this case requires the whole series of graph minor papers and the structure theorem of [33].

## 1.5 Our main contributions – Unique linkage theorem

The analysis of Case 2 in the previous subsection can be reduced to the Unique Linkage Theorem without excessive difficulty. It is then the proof of the Unique Linkage Theorem that requires much of the graph minors machinery. In fact, in the corresponding argument for the Graph Minors Algorithm, this is the only place in the proof of correctness which requires the full structure theorem. Before stating the theorem, we give some notation.

A *linkage* is a graph where every component is a path (possibly trivial). The *order* of the linkage is the number of components. In slightly sloppy notation, we will use $P \in \mathcal{P}$ to refer to a component $P$ of the linkage $\mathcal{P}$. Two linkages $\mathcal{P}$ and $\mathcal{P}'$ are *equivalent* if they have the same order and for every component $P$ of $\mathcal{P}$, there exists a component $P'$ of $\mathcal{P}'$ such that $P$ and $P'$ have the same endpoints. We say a linkage $\mathcal{P}$ in a graph $G$ is *unique* if for all linkages $\mathcal{P}'$ in $G$ equivalent to $\mathcal{P}$, we have that $V(\mathcal{P}') = V(\mathcal{P})$.

We are now ready to state the theorem, which is the main result of Graph Minors XXI [34].

**Theorem 1** (**The Unique Linkage Theorem [34]**). *For all $k \geq 1$, there exists a value $w(k)$ such that the following holds. Let $\mathcal{P}$ be a linkage of order $k$ in a graph $G$ with $V(G) = V(\mathcal{P})$. If $\mathcal{P}$ is unique, then the tree-width of $G$ is at most $w(k)$.*

The current proof given by Robertson and Seymour [34] needs the full power of the graph minor structure theorem, but they predicted that there exists a simpler proof avoiding the use of the Graph Minor structure theorem. Our main contribution is to confirm that they are right– we provide such a short proof. Our proof less is than 25 pages, and gives rise to an explicit bound for the tree-width $w(k)$, while the original algorithm does not.

We now mention several consequences of our new shorter proof. First, we clarify how the unique linkage theorem implies that the vertex $v$ in Case 2 is irrelevant. This was easy to prove for Case 2.1. The formal argument is given in Sections 5 and 6 in [31]. We are left with Case 2.2. The main result in [35] says that the existence of the irrelevant vertex in Case 2.2 can be reduced to the unique linkage problem. Let us observe that the arguments in [35] does NOT use the graph minor structure theorem. It is totally self-contained. Our proof of the Unique Linkage Theorem currently uses several tools from [29] for graphs embedded on surfaces of bounded genus (again, these tools do not depend on the structure theorem). Thus together with [35] and [29], our proof of the Unique Linkage Theorem provides a proof of the correctness of the $k$-disjoint paths algorithm which avoids the use of the graph minor structure theorem. At the moment, we believe that we also have a much shorter proof of the main result in [35] and the aspects of [29] which we use. This would lead to a correspondingly short, self-contained proof of the $k$-vertex disjoint paths algorithm.

Second, when we consider instead the $k$-edge disjoint paths problem, we are able to avoid the need for the work of [35]. This allows us to give a self-contained argument for the proof of correctness of the $k$-edge disjoint paths problem. We present the argument in the next subsection.

Finally, one of the original applications of the Unique Linkage Theorem is to verify the "intertwining conjecture" of Lovász [21] and Milgram and Unger [24]. The conjecture states that for every two graphs $G_1$ and $G_2$, there is a finite list $H_1, \ldots, H_n$ of graphs, such that $G$ topologically contains both $G_1$ and $G_2$ if and only if it topologically contains one of $H_1, \ldots, H_n$ ($G$ *topologically* contains $H$ if some subgraph of $G$ is isomorphic to a subdivision of $H$). A proof of this conjecture follows from the unique linkage theorem, as proved in [34]. But our proof, together with the proof in Section 11 of [34] gives rise to a short self-contained proof of this conjecture, which is, we believe, of independent interest. As pointed out in [34], our proof yields an algorithm that given two graphs $G_1$ and $G_2$, computes $H_1, \ldots, H_n$ above.

We conclude with a few words on possible applications of our new proof. Kernelization is a technique for creating algorithms for fixed-parameter tractable problems. This concept has attracted recent interest within the framework of parameterized complexity. See, for example, [3]. The approach is based on the observation that a problem is fixed-parameter tractable if and only if it is kernelizable and decidable. The idea of kernelization is to reduce the size of the input $X$ to a function of $k$ in polynomial time. When the input is bounded by $k$, we can use any exponential time algorithm, for example brute-force search, to find a solution of the problem. A basic technique in kernelization arguments is to find an "irrelevant" vertex for the problem, and reduce the size of the input. This is exactly what we do for the disjoint paths problem, hence we hope that our new short proof might yield new technical methods in this line of inquiry.

Algorithms for $H$-minor-free graphs for a fixed graph $H$ have been studied extensively; see e.g. [4, 5, 6, 12]. In particular, it is generally believed that algorithms for planar graphs can often be generalized to $H$-minor-free graphs for any fixed $H$. Results from graph minors, and the unique linkage theorem in particular, are essential for these arguments. For example, the topological embedding algorithms given in [14, 15, 16, 17, 23] partially depend on the unique linkage theorem. Also, linear time algorithms for the disjoint paths problem when an input graph is planar [26] or an input graph is bounded genus [9, 20] heavily depend on the unique linkage theorem. Thus we anticipate that our new proof will have further applications along these lines.

## 1.6 The edge-disjoint paths problem

Using our new proof of Unique Linkage Theorem, we give a short proof of correctness for the *$k$-edge disjoint paths problem.*

**Input**: A graph $G$ with $n$ vertices and $m$ edges, $k$ pairs of vertices $(s_i, t_i)$, called *terminals*, $i = 1, \ldots, k$, in $G$.

**Output** : Edge-disjoint paths $P_1, P_2, \ldots, P_k$ in $G$ such that $P_i$ joins $s_i$ and $t_i$ for $i = 1, 2, \ldots, k$.

We assume that $k$ is fixed. We will need the following definitions. For a vertex set $X$ in a graph $G = (V, E)$, let $\delta(X)$ be the set of edges between $X$ and $V \setminus X$. For a graph $G = (V, E)$, its *line graph* $L(G)$ is the graph whose vertex set is $E$ such that two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in $G$. To simplify the description, when we consider the line graph of a graph with terminals, we assume that exactly one edge is incident to each terminal by adding a new terminal and a single edge to $G$. Let $\tilde{s}_1, \ldots, \tilde{s}_k, \tilde{t}_1, \ldots, \tilde{t}_k$ be the edges incident with the terminals $s_1, \ldots, s_k, t_1, \ldots, t_k$ in $G$, respectively. Then, one can see that the edge-disjoint paths problem in $G$ with respect to the terminals $s_1, \ldots, s_k, t_1, \ldots, t_k$ is equivalent to the vertex-disjoint paths problem in $L(G)$ with respect to the terminals $\tilde{s}_1, \ldots, \tilde{s}_k, \tilde{t}_1, \ldots, \tilde{t}_k$.

As proved in [19], an instance of the $k$ edge-disjoint paths problem can be reduced to an instance satisfying the following conditions:

(R1) All vertices have degree at most $2k - 1$.

(R2) $G$ has no vertex set $X$ such that $|X| \geq 2$, $X$ contains no terminals, and $|\delta(X)| \leq 3$.

(R3) $G$ and $L(G)$ has no clique minor of size $3k$.

We call these operations *simple reductions*. Although it is easy to find a vertex of high degree (as in (R1)) and a $\leq 3$-edge-cut in a given graph (as in (R2)), it is not easy to find a large clique minor.

The following theorem, which is the main result in [19], characterizes the instances of the edge-disjoint paths problem, and shows a way to find a large clique minor.

**Theorem 2.** *For any instance of the $k$-edge-disjoint paths problem and for any integer $h \geq 2$, there exists an integer $f(k, h)$ such that one of the following holds:*

*(A) The instance violates at least one of (R1), (R2), and (R3). That is, one of the simple reductions can be applied to the instance.*

*(B) The input graph has tree-width at most $f(k, h)$.*

*(C) The input graph contains a wall $W$ of size $h$ with the outer face boundary $C$ with the following properties:*

*(a) $G - C$ consists of two parts $X$ and $Y$ such that $X \cup C$ contains the whole wall $W$.*

*(b) Every vertex of $X \cup C$ has degree at most three.*

*(c) $X \cup C$ does not contain any terminal.*

*(d) $X \cup C$ can be embedded in a plane with the outer face boundary $C$.*

*We can find one of (A), (B) and (C) in $O(m)$ time. Furthermore, if the instance satisfies (R1) and (R2), but does not satisfy (B) and (C), then we can find a clique minor of size $3k$ in $G$ or $L(G)$ in $O(m)$ time (For definitions of the tree-width and the wall, we refer the reader to the appendix).*

Having Theorem 2, we are ready to describe our $O(mn)$ time algorithm for the edge disjoint path problem more precisely. The algorithm below has appeared in [19], but for the completeness, we include the whole algorithm. We set $h = 4w(k)+4$, where $w(k)$ is the value given by Theorem 1.

**Algorithm for the edge-disjoint paths problem**

**Step 1.** We first apply Theorem 2. If (A) in Theorem 2 occurs, we apply a simple reduction as in (A) and recurse on a smaller graph. If (B) occurs, we apply the standard dynamic programming argument [1, 2]. Thus we may assume outcome (C).

**Step 2.** If (C) happens, it is possible to throw away a vertex $v$ (*irrelevant vertex*) in the deep inside the wall $W$ if $h$ is large enough (i.e, the vertex in the middle brick of $W$).

We then recursively apply our algorithm to $G - v$. Since Theorem 2 can be done in $O(m)$ time, the whole algorithm runs in $O(mn)$ time (this improves the time complexity of [31] that gives an $O(m^3)$ algorithm for the edge-disjoint paths problem).

**Correctness of the Algorithm**

For the correctness of the algorithm, it suffices to prove that $v$ is an irrelevant vertex in Step 2. We now give a proof, which is very similar to that given in [35], Theorem (3.1). We shall essentially reduce the correctness of the algorithm to the unique linkage theorem.

It is easy to see that if $G$ does not have the desired $k$ edge-disjoint paths, then $G - v$ does not have them either. Thus it remains to show that if the desired $k$ edge-disjoint paths exist in $G$, then $G - v$ has them as well. Let $G'$ be the line graph $L(G)$. We begin with the following:

(1) The line graph of $X \cup C$ described in (C) of Theorem 2 is still a plane graph.

This is because each vertex in $X \cup C$ has degree at most three in $X \cup C$. Hereafter, let $H$ be the plane subgraph of $G'$ induced by $X \cup C$.

Let $C_1, \ldots, C_s$ be disjoint cycles in the plane graph $H$. Let $D_i$ be the disc in the plane with boundary $C_i$. We say that they are *concentric* if we have the property that $D_s \subseteq \cdots \subseteq D_1$. Let $C_1, \ldots, C_{h/2}$ be concentric cycles in $H$ and $\mathcal{P} = \{P_1, \ldots, P_k\}$ be a linkage in $G'$. Note that since $X \cup C$ contains a wall of height $h$, it follows that $H$ also contains these $h/2$ concentric cycles.

We assume that the vertices of $G'$ that correspond to the edges incident with $v$ in Step 2 are contained in $D_{h/2} - C_{h/2}$ (again such a choice is possible by the above remark). Let $M = C_1 \cup \cdots \cup C_h$. We only need to prove is the following:

(2) Suppose $M$ exists in $G'$. Then the desired $k$ vertex-disjoint paths in $G'$ exist such that the vertices of $G'$ that correspond to the edges incident with $v$ in $G$ are not in any of the paths.

This will clearly suffice to complete the proof of the theorem. We prove (2) by induction on the number of vertices of $G'$. Note that we do not preserve line graph in the inductive step, i.e, when we make a smaller graph and apply induction, it may not be the line graph of some graph. We only require that our graph is contained in $H$ as a subgraph, i.e, $h/2$ concentric cycles in a subgraph of the plane graph $H$.

Proceeding, if there is a vertex $u$ that is not in $M \cup \mathcal{P}$, then we can delete $u$ from $G'$, and apply induction to $G' - u$. Similarly, consider the case when there exists an edge $e$ that is in $C_i$, $i \leq t/2$, but one of the endpoints is not used in $\mathcal{P}$. We can contract $e$ and still preserves the existence of concentric cycles $C_1, \ldots, C_i/e, \ldots, C_{h/2}$ (and a plane subgraph $H/e$), unless $|C_i| = 3$. But if $|C_i| = 3$, then we can clearly reroute the paths in $\mathcal{P}$ so that they do not touch any vertex inside the disk $D_i$, except for the vertices in $C_i$, and so find a linkage avoiding the edges incident $v$. Thus, after contracting $e$, we can apply induction to the resulting graph. We conclude that $V(M \cup \mathcal{P}) = V(G')$.

Let $w(k)$ be the value given by Theorem 3. By a *dive* we mean a subpath of a path in $\mathcal{P}$ contained in the disc $D_1$ with both ends in $C_1$ and at least one vertex in $C_l$ for some $l \geq w(k) + 1$ We now claim the following:

(3) There are at most $w(k)$ dives.

If there is another linkage $\mathcal{P}'$ equivalent to $\mathcal{P}$ such that $|V(\mathcal{P}')| < |V(\mathcal{P})|$, then there is a vertex $u$ of $G'$ that is not in $\mathcal{P}'$. If $u$ is not in $M$, then we delete $u$ from $G'$, and apply induction to $G' - u$. Similarly, if $u$ is in $M$, then there is an edge $e$ with one endpoint $u$ in $M$. In this case, we contract $e$ as above. After contracting $e$, we can apply the inductive hypothesis to the resulting graph. Thus we may assume that $\mathcal{P}$ is a unique linkage.

We now use the unique linkage theorem to prove (3). Suppose for a contradiction that the linkage $\mathcal{P}$ contains at least $w(k) + 1$ dives. Then since $H$ is a plane subgraph of $G'$ and $M$ is contained in $H$, there are dives $P_1, P_2, \ldots, P_{w(k)+1}$ that are pairwise disjoint and all intersect $C_i$ for $i = 1, \ldots, w(k)+1$. This implies that $P_1, P_2, \ldots, P_{w(k)+1}$ all intersect each of $C_1, C_2, \ldots, C_{w(k+1)}$, and hence $C_1 \cup P_1, C_2 \cup P_2, \ldots, C_{w(k)+1} \cup P_{w(k)+1}$ is a "bramble" in $G'$ of "order" at least $w(k)+1$ (for the definition of the bramble, we refer the reader to [25]). By [25] the graph $G'$ has tree-width at least $w(k) + 1$, a contradiction to the unique linkage theorem. This proves (3).

We are now ready to finish the proof. We claim that no dive intersects $C_{2w(k)+1}$. The *depth* of a dive $P$ is the maximum index $i$ such that $P \cap C_i \neq \emptyset$. To see this, observe that if $P$ is a dive of depth $i$, then if $C_{i-1}$ does not intersect any path of $\mathcal{P}$, we can reroute the component of $\mathcal{P}$ containing $\mathcal{P}$ to avoid the vertex $P \cap C_i$. Thus, some component (other than the one containing $P$) of $\mathcal{P}$ intersects $C_{i-1}$. By planarity, it follows that there exists a dive of depth $i - 1$. Thus, if there exists a dive of depth $2w(k) + 2$, we see that there exist $w(k) + 1$ dives, a contradiction to (3).

If we assume that $h \geq 4w(k)+4$, we see that no component of $\mathcal{P}$ can intersect $C_{t/2}$. This completes the proof of (2), and the theorem. □

In the next section, we give an outline of our proof of the unique linkage theorem. To help the reader see how the proof goes, we shall give a short proof of the case $k = 2$.

## 2. OUTLINE OF THE PROOF OF THE UNIQUE LINKAGE THEOREM

In this section, we give an overview for our proof of the unique linkage theorem[1]. The proof proceeds by analyzing what we will call *traversing linkages*. Before we give the exact definition, we first give some intuition of what a traversing linkage is. Let $\mathcal{P}$ be a $k$-linkage. A linkage $\mathcal{Q}$ traverses $\mathcal{P}$ if when we follow the linkage $\mathcal{Q}$ from beginning to end, we intersect the linkage $\mathcal{P}$ repeatedly in a regular, uniform way. Moreover, these intersections are independent of each other in a sense. That is, the first intersections of $\mathcal{P}$ and $\mathcal{Q}$ are contained in a small subpath of $\mathcal{P}$, and $\mathcal{Q}$ never returns to that subpath. We reduce the proof of the Unique Linkage Theorem to showing the following theorem.

**Theorem 3.** *There exists functions $l(k)$ and $w(k)$ such that the following holds. Let $\mathcal{P}$ be a linkage of order $k$, and let $\mathcal{Q}$ be a linkage traversing $\mathcal{P}$ of order $w(k)$ and length $l(k)$. Then $\mathcal{P}$ is not unique in $\mathcal{P} \cup \mathcal{Q}$.*

Traversing linkages have two nice properties we use repeatedly in the proof of Theorem 3. First, the graph consists of the union of just two linkages, and so is dramatically simpler than the general graphs typically analyzed in the theory of graph minors. Second, there is an element of symmetry allowing us to move back and forth between analyzing first the linkage $\mathcal{P}$, and then the linkage $\mathcal{Q}$, and back again.

We now give the exact definition of a traversing linkage. We recall that a *ladder of length $t$* is a graph consisting of two paths of length $P_1$, $P_2$ with the vertices of $P_i$ equal to $v_1^i, \ldots, v_t^i$ for $i = 1, 2$ as well as edges of the form $v_j 1 v_j 2$ for $1 \leq j \leq t$.

**Definition 4.** Let $\mathcal{P}$ be a linkage. The linkage $\mathcal{Q}$ *traverses* $\mathcal{P}$ (or, equivalently, is a traversing linkage) if there exist disjoint subpaths $B_1, \ldots, B_l$ in $\mathcal{P}$ such that the following hold:

a. The linkage $\mathcal{Q}$ intersects $\mathcal{P}$ only in the subpaths $B_1, \ldots, B_l$, i.e. $V(\mathcal{Q}) \cap V(\mathcal{P}) \subseteq \bigcup_1^l V(B_i)$.

b. For all $Q \in \mathcal{Q}$ and $1 \leq i \leq l$, $Q \cap B_i$ is a (possibly trivial) subpath of $B_i$.

c. For every element $Q \in \mathcal{Q}$, we may traverse the path $Q$ from one end to the other, we encounter the paths $B_1, B_2, \ldots, B_l$ in that order.

d. If we look at the $Z$ the set of subpaths of $\mathcal{Q}$ with one end in $B_i$ and another end in $B_{i+1}$ for $1 \leq i \leq l-1$, then $Z \cup B_i \cup B_{i+1}$ forms a subdivided ladder after possibly deleting vertices of degree one.

The paths $B_1, B_2, \ldots, B_l$ are called the *basis subpaths* of the traversing linkage $\mathcal{Q}$. The value $l$ is the *length* of the traversing linkage $\mathcal{Q}$. Again, the order of the traversing linkage is the number of components. Fix labels $s_P$, $t_P$ to the endpoints of every component $P \in \mathcal{P}$. If we consider the ladder in d, there are two distinct possibilities. Let $P$ and $P'$ be the components of $\mathcal{P}$ containing $B_i$ and $B_{i+1}$, respectively. We say that $\mathcal{Q}$ *twists between $B_i$ and $B_{i+1}$* if for all $j$, the $j^{th}$ component of $Z$ we intersect traveling $P$ from $s_P$ to $t_P$ is the $(w-j+1)^{th}$ when traversing $P'$ from $s_{P'}$ to $t_{P'}$ (where $w$ is the order of $\mathcal{Q}$).

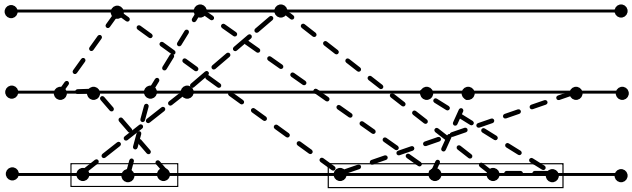

**Figure 1: The dashed linkage is a traversing linkage of order three and length five traversing the solid linkage. The first and fourth basis subpaths are indicated with boxes. The dashed linkage twists between the first and second basis subpaths.**

Observe, that if we swap the labels $s_P$ and $t_P$ on a component $P \in \mathcal{P}$, then if we consider some basis subpath $B_i$ contained in $P$, if $\mathcal{Q}$ twists between $B_i$ and $B_{i+1}$ before the swap, then $\mathcal{Q}$ will not twist between the two basis subpaths after the swap (and vice versa: if $\mathcal{Q}$ does not twist before the swap, then it will twist after the swap).

We briefly describe now how we reduce the Unique Linkage Theorem to the proof of Theorem 3. The analysis is somewhat similar to the proof of the edge disjoint version of the disjoint paths problem. We proceed in two basic steps. First, we pick a prospective counter-example to the Unique Linkage Theorem: a linkage $\mathcal{P}$ contained in a graph $G$ with $V(G) = V(\mathcal{P})$ such that $G$ does not contain an equivalent linkage on fewer vertices. Moreover, we make the assumption that the tree width of the graph is huge. We first show that such a counterexample $G$ cannot contain a large clique minor, and then using what we call the Weak Structure Theorem, we show that there exists a large planar subgraph $H$ containing a huge wall such that the linkage $\mathcal{P}$ interacts with $H$ planarly. In other words, even if there exist vertices with neighbors in the center of $H$, the components of $\mathcal{P}$ intersect $H$ in a way that always respects the planar embedding of $H$. We then, as in the proof of the edge disjoint version, take a large number of concentric cycles such that they intersect the linkage $\mathcal{P}$ in a clean way. The concentric cycles as they travel through the linkage $\mathcal{P}$ will then provide the traversing linkage $\mathcal{Q}$.

The proof of Theorem 3 will be the main work in our proof of the Unique Linkage Theorem. The remainder of this section will be devoted to a brief outline of the proof of Theorem 3.

The proof of Theorem 3 proceeds by finding many sublinkages in $\mathcal{Q}$ forming what we will call $\mathcal{Q}$-bumps. Let $\mathcal{P}$ be a linkage and $\mathcal{Q}$ be a traversing linkage of order $w$. Let $B_1, \ldots, B_l$ be the basis subpaths. A *$\mathcal{Q}$-bump* is a sublinkage $\overline{\mathcal{Q}}$ of $\mathcal{Q}$ of order $w$ such that there exist indices $i$ and $i'$ and a path $P \in \mathcal{P}$ such that

a. every component of $\overline{\mathcal{Q}}$ has one endpoint in $B_i$ and one endpoint in $B_{i'}$ and no internal vertex in $P$, and

b. $B_i$ and $B_{i'}$ are both contained in $P$.

$\mathcal{Q}$-bumps can be thought of as a cylindrical set of subpaths wrapping around a sublinkage of the linkage $\mathcal{P}$. A $\mathcal{Q}$ bump allows one to reroute the linkage $\mathcal{P}$ - not to find an equivalent

[1]A full version of this paper is available at `http://reseach.nii.ac.jp/~k_keniti/uniquelink.pdf`

linkage - but rather to cyclically shift by one some subset of the paths. We make this more explicit in the following observation.

**Observation 5.** *Let $\mathcal{P}$ be a linkage of order $k$ with components $P_i$ for $1 \leq i \leq k$. Let $s_i$ and $t_i$ be the endpoints of $P_i$ for $1 \leq i \leq k$. Let $\mathcal{Q}$ be a traversing linkage of order $k+1$ with basis subpaths $B_1, \ldots, B_l$. Let $\overline{\mathcal{Q}}$ be a $\mathcal{Q}$-bump of length $l'+1$ with basis subpaths $\overline{B}_1, \ldots, \overline{B}_{l'+1}$ satisfying the following properties.*

*i. Assume $\overline{B}_i$ is contained in $P_i$ for $1 \leq i \leq l'$. Specifically, note $\overline{B}_i$ and $\overline{B}_{i'}$ are contained in distinct components of $\mathcal{P}$ for $1 \leq i < i' \leq l'$.*

*ii. For all $i$, $1 \leq i < l'+1$, $\mathcal{Q}$ does not twist between $B_i$ and $B_{i+1}$.*

*Then $\mathcal{P} \cup \overline{\mathcal{Q}}$ contains disjoint paths $P'_1, \ldots, P'_k$ such that the endpoints of $P'_i$ are $s_i$ and $t_{i+1}$ for $1 \leq i \leq l'$ (taken modulo $l'$) and the ends of $P_i$ are $s_i$ and $t_i$ for $i > l'$. Moreover, the paths $P'_1, \ldots, P'_k$ can be chosen to avoid some vertex of $\mathcal{P}$.*

We illustrate the observation in Figure 2.

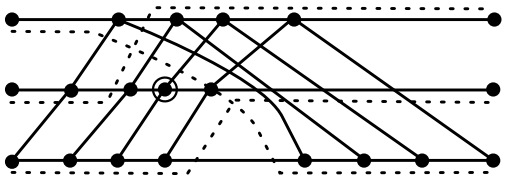

**Figure 2: A bump of order 4 traversing a linkage of order 3. The rerouted paths guaranteed by Observation 5 are indicated as dotted paths. Note that the circled vertex is avoided by the new linkage.**

Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be two vertex disjoint $\mathcal{Q}$-bumps, and assume that there exists a component $P$ of $\mathcal{P}$ containing all the endpoints of $\mathcal{R}_1$ and $\mathcal{R}_2$. Let $\pi(i)$ and $\sigma(i)$ be values, for $i = 1, 2$ such that $\mathcal{R}_i$ has endpoints in $B_{\pi(i)}$ and $B_{\sigma(i)}$. Consider the subpath of $P_i$ of $P$ connecting the ends of $R_i$, i.e. let $P_i$ be the minimal subpath of $P$ containing $B_{\pi(i)}$ and $B_{\sigma(i)}$ for $i = 1, 2$. Then $\mathcal{R}_1$ and $\mathcal{R}_2$ are *independent* if $P_1$ and $P_2$ are disjoint.

We now have everything in place to outline the argument for the proof of Theorem 3. Let $\mathcal{P}$ be our linkage of order $k$ which we would like to reroute, and let $\mathcal{Q}$ be a *very* long traversing linkage of *huge* order. Fix a path $P_1$ in the linkage $\mathcal{P}$.

First, we observe that for some component $P_1$ of $\mathcal{P}$, the linkage $\mathcal{Q}$ must return frequently to the path $P_1$.

If there existed many disjoint independent $\mathcal{Q}$-bumps with ends in $P_1$, it would be relatively easy to ensure that some subset large satisfies the necessary properties to apply Observation 5. Then using at most $k$ bumps satisfying the conditions of Observation 5, we repeatedly rotate and find an equivalent linkage avoiding some vertex.

The difficulty is how to find many disjoint independent $\mathcal{Q}$-bumps. We switch perspectives at this point and consider how the components of $\mathcal{P}$ intersect with $\mathcal{Q}$. The path $P_1$ repeatedly crosses the linkage $\mathcal{Q}$, always hitting the paths of $\mathcal{Q}$ in order. Then there are two possible cases. First, it is possible that some subpath $P'_1$ of $P_1$ intersects $\mathcal{Q}$ in a somewhat regular manner, yielding many disjoint, independent $\mathcal{Q}$-bumps on $P'_1$. In this case, we apply Observation 5 to find a linkage equivalent to $\mathcal{P}$ avoiding some vertex. Alternatively, no such path $P'_1$ exists and the path $P_1$ intersects $\mathcal{Q}$ in a more complex fashion. In this case we are able to find a large complete minor in $\mathcal{P} \cup \mathcal{Q}$. As we have already discussed, a large clique minor allows us to find an equivalent linkage avoiding some vertex of $\mathcal{P}$. In each case, we find an equivalent linkage avoiding some vertex of $\mathcal{P}$, completing the proof.

We expand, for a moment, on these ideas for interested readers with some familiarity with the graph minors techniques. We return to a more general discussion below. Let $\mathcal{P}$ be a linkage of order $k$ and $\mathcal{Q}$ be a traversing linkage of order $w$ and length $l$. Let the basis subpaths of $\mathcal{Q}$ be $B_1, B_2, \ldots, B_l$.

We contract all the edges incident a vertex of degree at most two, as well as edges of $E(\mathcal{Q}) \cap E(\mathcal{P})$. Thus we may assume that:

1. there are no edges contained in $\mathcal{Q} \cap \mathcal{P}$, and

2. there are no vertices in $V(\mathcal{Q}) \setminus V(\mathcal{P})$.

We can fix a labeling $Q_1, Q_2, \ldots, Q_w$ of the components of $\mathcal{Q}$ of order $w$, so that every component $P \in \mathcal{P}$ satisfies the following. The path $P$ can be decomposed into subpaths $R_1, \ldots, R_t$ and edges $e_1, \ldots, e_{t-1}$ that are pairwise disjoint so that $e_i$ connects the ends of $R_i$ and $R_{i+1}$ and each $R_i$ has one end in $Q_1$, the other end in $Q_w$, and intersects the paths of $\mathcal{Q}$ in order, i.e. $Q_1, Q_2, \ldots, Q_w$. (In fact, the paths $R_i$ are the basis subpaths of $\mathcal{Q}$ on $P$, ordered by traversing $P$).

If we look at the graph formed by $P \cup \mathcal{Q}$, we see that $\mathcal{Q} \cup \bigcup_1^t R_i$ forms a subdivision $W$ of the $(w \times t)$-grid. The horizontal paths of the grid are the $Q_i$ for $1 \leq i \leq w$, and the vertical paths are the $R_i$, $1 \leq i \leq t$. The edges $e_i$ form a matching with the ends contained in $Q_1 \cup Q_w$. We keep the grid $W$ aside, and for the rest of the proof focus on the edges $e_i$. As we described above, there are essentially two cases. If these edges are relatively well behaved, all but a small number of them can be embedded in a low genus surface. In this case, this will allow us to reroute the linkage $\mathcal{P}$ avoiding some vertex using techniques of Robertson and Seymour for the disjoint paths problem in the bounded genus case [29]. Alternatively, the edges $e_i$ are not tame. In this case, we will find a large clique minor. Then, again we see that $\mathcal{P}$ can again be rerouted to avoid some vertex $v$.

The edges $e_i$, together with the outer face boundary of $W$, comprise a *society*, one of the key topics in Graph Minors Theory [30]. Our proof builds on results in [30], and extending them in such a way that the outcomes include "genus addition", i.e, a handle addition and a crosscap addition. We further adapt some ideas in [32, 33] to grow a graph on the surface with large representativity. This process stops when we have a huge clique minor, because, as mentioned above, if there is a huge clique minor, we can reroute the linkage $\mathcal{P}$ to avoid some vertex. We reiterate that our proof does not need most of the heavy machinery in Graph Minor theory. This is for several reasons. First, because our society consists of only a matching, the analysis is simplified. Second, we do not have to worry about global connectivity issues as the society vertices are the vertices of the outer ring

of a grid. And, finally, certain degenerate cases will allow us to easily find many disjoint $\mathcal{Q}$-bumps, an outcome not available in the general graph minors arguments. This final point will allow us to evade the topic of "vortices", a major savings in time and effort. Further ingredients of the graph minors series which we are able to avoid include, "embedding up to 3-separations", "tangle, respectful tangle" etc.

We return now to a more general overview of the proof. A technicality we ignored in this outline is the following. Given that we find many disjoint independent $\mathcal{Q}$-bumps on the subpath $P_1'$ of $P_1$, how do use the bumps to reroute the linkage? We do so by *splitting* on edges. Given an edge $e$ in a linkage $\mathcal{P}$, we say that the linkage $\mathcal{P} - e$ is obtained from $\mathcal{P}$ by splitting $\mathcal{P}$ on $e$. We note that the property of being a unique linkage is preserved upon splitting a linkage on a given edge:

**Observation 6.** *Let $\mathcal{P}$ be a unique linkage in a graph $G$, and let $\overline{\mathcal{P}}$ be obtained from splitting $\mathcal{P}$ on some edge $e$. Then $\overline{\mathcal{P}}$ is a unique linkage in $G$. Moreover, if $\mathcal{Q}$ is a traversing linkage of $\mathcal{P}$ of length $l$, with basis subpaths $B_1, B_2, \ldots, B_l$, then if $e \notin E(B_i)$ for all $1 \leq i \leq l$, then $\mathcal{Q}$ is a traversing sublinkage of $\overline{\mathcal{P}}$.*

Thus we perform possibly two edge splits on $\mathcal{P}$ to obtain a new linkage with a component equal to $P_1'$. Given that we have many disjoint, independent $\mathcal{Q}$ bumps attaching to $P_1'$, we contradict that the new linkage is unique, and by the observation, that the original linkage $\mathcal{P}$ is unique.

In conclusion, we give a complete proof of the $k = 2$ case of Theorem 3. Robertson and Seymour [34] observe that this can be easily shown directly; however, we give a proof using traversing linkages in the hopes that it further illustrates the tools and techniques of the main result.

**Theorem 7.** *Let $\mathcal{P}$ be a linkage of order 2. Let $\mathcal{Q}$ be a traversing linkage of order five and length 33. Then there exists a vertex $v \in V(\mathcal{P})$ such that $(\mathcal{P} \cup \mathcal{Q}) - v$ contains a linkage $\mathcal{P}'$ equivalent to $\mathcal{P}$.*

PROOF. Let the components of $\mathcal{P}$ be $P_1$ and $P_2$, and label the ends of $P_i$ $s_i$ and $t_i$ for $i = 1, 2$. Let the basis subpaths of $\mathcal{Q}$ be $B_1, B_2, \ldots, B_{33}$. We assume, to reach a contradiction, that there do not exist paths $P_1'$, $P_2'$ that avoid some vertex of $\mathcal{P}$ such that the endpoints of $P_i'$ are $s_i$ and $t_i$.

The linkage $\mathcal{Q}$ repeatedly passes back and forth between $P_1$ and $P_2$. To simplify the picture somewhat, we consider the following auxiliary graph $H$ with vertices equal to the set of basis subpaths $B_i$ for $1 \leq i \leq 33$ and two vertices $B_i$ and $B_j$ connected by an edge if either there is a subpath of $P_i$ connecting them avoiding all other basis subpaths, or if $|j - i| = 1$. It follows that $E(H)$ is comprised of three edge disjoint paths: one for each of the paths $P_i$ and one for the linkage $\mathcal{Q}$. The edges of $H$ of the form $B_iB_{i+1}$ have two distinct types - the linkage $\mathcal{Q}$ can either twist between $B_i$ and $B_{i+1}$ or not. We will refer to the edges $B_iB_{i+1}$ where $\mathcal{Q}$ twists as the *odd* edges of $H$; every other edge of $H$ will be called *even*. If $\Sigma$ is the set of odd edges, then by swapping the labels $s_i$ and $t_i$, the resulting set of odd edges is $\Sigma \triangle X$ where $\triangle$ denotes the symmetric difference and $X$ is the set of all edges of the form $B_iB_{i+1}$.

First, observe that there does not exist an index $i$ such both the edge $B_{i-1}B_i$ and $B_iB_{i+1}$ are even in $H$. Otherwise, we can find paths $P_1'$ and $P_2'$ equivalent to $\mathcal{P}$ avoiding an internal vertex of $B_i$. Similarly, there does not exist an index $i$ such that $\mathcal{Q}$ does twist between both $B_{i-1}$ and $B_i$ and between $B_i$ and $B_{i+1}$. This is because we could swap the labels on the ends $P_1$ so that $\mathcal{Q}$ does not twist between $B_{i-1}$ and $B_i$ and between $B_i$ and $B_{i+1}$. If we let $R$ be the path in $H$ of consisting of the edges of the form $B_iB_{i+1}$ for $1 \leq i \leq 32$, it follows that the edges of $R$ alternate between edges in $\Sigma$ and edges not in $\Sigma$.

Also, observe that for indices $i$ and $j$ such that both the edges $B_iB_{i+1}$ and $B_jB_{j+1}$ are not in $\Sigma$, we have that the edges do not "cross" in $H$. That is, if $B_i$ and $B_j$ are both contained $P_1$, say, and occur on $P_1$ in that order when traversing from $s_1$ to $t_1$, then $B_{i+1}$ occurs before $B_{j+1}$ on $P_2$ when traversing from $s_2$ to $t_2$. Otherwise, we would be able to find an equivalent linkage avoiding some vertex of $\mathcal{P}$. Similarly, if $i$ and $j$ are two indices such that both the edges $B_iB_{i+1}$ and $B_jB_{j+1}$ are in $\Sigma$, and if both $B_i$ and $B_j$ are contained in $P_1$ in that order, then it follows that $B_{j+1}$ and $B_{i+1}$ occur on that order when traversing $P_2$ from $s_2$ to $t_2$.

After possibly swapping the labels $s_i$ and $t_i$ for one or possibly both values of $i$, we see by examining the graph $H$ that the following must hold for $\mathcal{Q} \cup \mathcal{P}$. Traversing $P_1$ from $s_1$ to $t_1$, we see $B_2, B_6, \ldots, B_{26}, B_{30}, B_{32}, B_{28}, \ldots, B_8, B_4$ in that order, and traversing $P_2$ from $s_2$ to $t_2$, we see $B_1, B_5, \ldots, B_{29}$, $B_{33}, B_{31}, B_{27}, \ldots, B_7, B_3$ in that order. Moreover, $\mathcal{Q}$ twists between $B_i$ and $B_{i+1}$ if and only if $i$ is even. Let $e_1$ be an edge of $P_1$ separating $B_{30}$ from $B_{32}$, and let $e_2$ be an edge of $P_2$ separating $B_{33}$ from $B_{31}$. If we split the linkage $\mathcal{P}$ on $e_1$ and $e_2$, we have a linkage $\mathcal{P}'$ of order four. Moreover, by appropriately choosing the labels for the endpoints of components of $\mathcal{P}'$, we may assume that $\mathcal{Q}$ does not twist between any two basis subpaths in $\mathcal{P}'$. We label the components of $\mathcal{P}'$ as $P_i'$ for $1 \leq i \leq 4$ such that the endpoints of $\mathcal{Q}$ are contained in $P_1'$.

By examination, we see that there exist four disjoint independent $\mathcal{Q}$-bumps with endpoints in $P_1'$ satisfying *i.* and *ii.* in Observation 5. Moreover, if we let the endpoints of $P_i'$ be $s_i'$ and $t_i'$, then for each such bump, the rerouting guaranteed by Observation 5 results in paths with endpoints $s_i'$ and $t_{i+1}'$. By using all four re-routings, we see that we wrap around and find a linkage $\overline{\mathcal{P}}$ equivalent to $\mathcal{P}'$ avoiding some vertex of $\mathcal{P}'$. Then Observation 6 implies a contradiction to our choice of $\mathcal{P}$ to be a unique linkage, proving the claim.

# 3. REFERENCES


[1] S. Arnborg and A. Proskurowski, Linear time algorithms for NP-hard problems restricted to partial $k$-trees, *Discrete Appl. Math.* **23** (1989), 11–24.

[2] H. L. Bodlaender, A linear-time algorithm for finding tree-decomposition of small treewidth, *SIAM J. Comput.* **25** (1996), 1305–1317.

[3] H. L. Bodlaender, F. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh and D. Thilikos, (Meta) kernelization, to appear in *50th Annual Symposium on Foundations of Computer Science (FOCS 2009).*

[4] E. D. Demaine, F. Fomin, M. Hajiaghayi, and D. Thilikos, Subexponential parameterized algorithms on bounded-genus graphs and $H$-minor-free graphs, *J. ACM* **52** (2005), 1–29.

[5] E. D. Demaine, M. Hajiaghayi, and K. Kawarabayashi, Algorithmic graph minor theory: Decomposition, approximation and coloring, *Proc. 46th Annual Symposium on Foundations of Computer Science (FOCS'05)*, 637–646, (2005).

[6] E. D. Demaine, M. Hajiaghayi, and B. Mohar, Approximation algorithms via contraction decomposition, *Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'07)*, 278–287, (2007).

[7] R. Diestel, Graph Theory, 3rd Edition, Springer, 2005.

[8] R.G. Downey and M.R. Fellows, Parameterized complexity, Springer-Verlag, 1999.

[9] Z. Dvorak, D. Kral and R. Thomas, Coloring triangle-free graphs on surfaces, *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2009, 120–129.

[10] M.R. Fellows and M. A. Langston, Nonconstructive tools for proving polynomial-time decidability, *J. ACM* **35**, (1988), 727–739.

[11] A. Frank, Packing paths, cuts and circuits – a survey, in *Paths, Flows and VLSI-Layout*, B. Korte, L. Lovász, H. J. Promel and A. Schrijver (Eds.), Springer-Verlag, Berlin, 1990, 49–100.

[12] M. Grohe, Local tree-width, excluded minors, and approximation algorithms. *Combinatorica* **23**, (2003), 613–632.

[13] R. M. Karp, On the computational complexity of combinatorial problems, *Networks* **5** (1975), 45–68.

[14] K. Kawarabayashi, Planarity allowing few error vertices in linear time, *50th Annual Symposium on Foundations of Computer Science (FOCS 2009)*, 639–648, (2009).

[15] K. Kawarabayashi and B. Reed, Computing crossing number in linear time, *Proc. 39th ACM Symposium on Theory of Computing (STOC'07)*, 382–390, (2007).

[16] K. Kawarabayashi and B. Mohar, Graph and Map Isomorphism and all polyhedral embeddings in linear time, *Proc. 40th ACM Symposium on Theory of Computing (STOC'08)*, 471–480, (2008).

[17] K. Kawarabayashi, B. Mohar and B. Reed, A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of bounded tree-width graphs, *Proc. 49th Annual Symposium on Foundations of Computer Science (FOCS'08)*, 771–780, (2008).

[18] K. Kawarabayashi, Y. Kobayashi and B. Reed, The disjoint paths problem in quaratic time, *submitted.*

[19] K. Kawarabayashi and Y. Kobayashi, The edge-disjoint paths problem for 4-edge-connected graphs and Eulerian graphs, *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 345–353, (2010).

[20] Y. Kobayashi and K. Kawarabayashi, Algorithms for finding an induced cycle in planar graphs and bounded genus graphs, *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1146–1155, (2009).

[21] L. Lovász, in *Combinatorics*, *Proc. Fifth Hungarian Combin. Colloq.*, Keszthely, 1976, Vol. II, p.1208, North-Holland, Amsterdam, 1978.

[22] J. F. Lynch, The equivalence of theorem proving and the interconnection problem, *ACM SIGDA Newsletter* **5** (1975), 31–65.

[23] D. Marx and I. Schlotter, Obtaining a planar graph by vertex deletion, *Proc. the 33rd Workshop on Graph-Theoretic Concepts in Computer Scienece*, 292–303, (2007).

[24] M. Milgram and P. Ungar, *Amer. Math. Monthly*, **85**, (1978), 664–668.

[25] B. Reed, Tree width and tangles: a new connectivity measure and some applications, in *"Surveys in Combinatorics, 1997 (London)"*, London Math. Soc. Lecture Note Ser. **241**, Cambridge Univ. Press, Cambridge, 87–162, (1997).

[26] B. Reed, N. Robertson, A. Schrijver and P. D. Seymour, Finding disjoint trees in planar graphs in linear time, Contemp. Math., 147, Amer. Math. Soc., Providenc, RI, 1993, 295–301.

[27] N. Robertson and P. D. Seymour, An outline of a disjoint paths algorithm, in *Paths, Flows, and VLSI-Layout*, B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver (Eds.), Springer-Verlag, Berlin, 1990, 267–292.

[28] N. Robertson and P. D. Seymour, Graph minors. VI. Disjoint paths across a disk, *J. Combin. Theory Ser. B*, **41** (1986), 115–138.

[29] N. Robertson and P. D. Seymour, Graph minors. VII. Disjoint paths on a surface, *J. Combin. Theory Ser. B*, **45** (1988), 212–254.

[30] N. Robertson and P. D. Seymour, Graph minors IX. Disjoint crossed paths, *J. Combin. Theory Ser. B*, **49** (1990), 40–77.

[31] N. Robertson and P. D. Seymour, Graph minors. XIII. The disjoint paths problem, *J. Combin. Theory Ser. B* **63** (1995), 65–110.

[32] N. Robertson and P. D. Seymour, Graph minors. XV. Giant Steps, *J. Combin. Theory Ser. B* **68** (1996), 112–148.

[33] N. Robertson and P. D. Seymour, Graph minors. XVI. Excluding a non-planar graph, *J. Combin. Theory Ser. B* **89** (2003), 43–76.

[34] N. Robertson and P. D. Seymour, Graph minors. XXI. Graphs with unique linkages, *J. Combin. Theory Ser. B* **99** (2009), 583–616.

[35] N. Robertson and P. D. Seymour, Graph minors. XXII. Irrelevant vertices in linkage problems, to appear in *J. Combin. Theory Ser. B.*

[36] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency, number 24 in Algorithm and Combinatorics, Springer Verlag, 2003.