Connecting Terminals and 2-Disjoint Connected Subgraphs

J.A. Telle[1] and Y. Villanger[1]

1 **Department of Informatics, University of Bergen, Bergen, Norway,**
`telle@ii.uib.no, yngvev@ii.uib.no`

Abstract

Given a graph $G = (V, E)$ and a set of terminal vertices T we say that a superset S of T is T-connecting if S induces a connected graph, and S is minimal if no strict subset of S is T-connecting. In this paper we prove that there are at most $\binom{|V\setminus T|}{|T|-2} \cdot 3^{\frac{|V\setminus T|}{3}}$ minimal T-connecting sets when $|T| \leq n/3$ and that these can be enumerated within a polynomial factor of this bound. This generalizes the well-known algorithm for enumerating all induced paths between a pair of vertices, corresponding to the case $|T| = 2$. We apply our enumeration algorithm to solve the 2-Disjoint Connected Subgraphs problem in time $O^*(1.7804^n)$, improving on the recent $O^*(1.933^n)$ algorithm of Cygan et al. 2012 LATIN paper.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

Keywords and phrases Keywords: Vertex set enumeration, Connected sets, Exact algorithms

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

The listing of all inclusion minimal combinatorial objects satisfying a certain property is a standard approach to solving certain NP-hard problems exactly. Some examples are the algorithms for Minimum Dominating Set in time $O^*(1.7159^n)$ [3], for Feedback Vertex Set in time $O^*(1.7548^n)$ [2], and for Minimal Separators in time $O^*(1.6181)$ [4].

This is an approach that usually requires little in the way of correctness arguments. For example, in the minimum dominating set problem it is obvious that a dominating set of minimum cardinality is also an inclusion minimal dominating set. The main task in this approach is to firstly enumerate the inclusion minimal objects, preferably by an algorithm whose runtime is within a polynomial factor of the number of such objects, and secondly to provide a good upper bound on the number of objects. Probably the most famous example is the polynomial delay enumeration algorithm for Maximum independent set [6] where there are matching upper and lower bounds on the number of objects [7].

Another case with matching upper and lower bounds is the well-known $O^*(3^{\frac{n-2}{3}})$ time algorithm enumerating all induced paths between two fixed vertices u and v in an n-vertex graph [1]. In this paper we consider some generalizations of this fundamental graph problem. We first generalize to the enumeration of induced paths starting in v and ending in a vertex from a given set R, with no intermediate vertices in R. The algorithm we give for this generalization will be optimal, up to polynomial factors. Given a subset of vertices T let us

[1] We have not been able to find a proof of this algorithm in the literature. The graph in Figure 1, with $|R| = 1$, shows optimality of the algorithm, up to polynomial factors.

Conference title on which this volume is based on.
Editors: Billy Editor, Bill Editors; pp. 1–10
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

say that a superset S of T is T-connecting if S induces a connected graph, and that S is minimal T-connecting if no strict subset of S is T-connecting. Our main generalization is the following enumeration task:

Enumeration of Minimal T-Connecting Sets
Input: A graph $G = (V, E)$ and a set $T \subseteq V$.
Output: All minimal T-connecting sets.

Note that for the case $|T| = 2$ the minimal T-connecting sets are in 1-1 correspondence with the set of induced paths between the two vertices of T. We give an algorithm for Enumeration of Minimal T-Connecting Sets with runtime $O^*(\binom{n-|T|}{|T|-2} \cdot 3^{\frac{n-|T|}{3}})$ where $|T| \leq n/3$. For $|T| > n/3$ a trivial $O^*(2^{n-|T|})$ brute force enumeration can be used. We apply this enumeration algorithm to solve the following problem:

2-Disjoint Connected Subgraphs
Input: A connected graph $G = (V, E)$ and two disjoint subsets of terminal vertices $Z_1, Z_2 \subseteq V$.
Question: Does there exist a partition A_1, A_2 of V, with $Z_1 \subseteq A_1, Z_2 \subseteq A_2$ and $G[A_1]$, $G[A_2]$ both connected?

The general version of this problem with an arbitrary number of sets was used as one of the tools in the result of Robertson and Seymour showing that Minor containment can be solved in polynomial time for every fixed pattern graph H [10]. We require the input graph to be connected since otherwise it is easy to reduce the problem to a connected component.

Let us look at some previous work on this problem. Motivated by an application in computational geometry, Gray et al [5] showed that 2-Disjoint Connected Subgraphs is NP-complete on planar graphs. van't Hof et al [11] showed that on general graphs it is NP-complete even when $|Z_1| = 2$ and also that it remains NP-complete on P_5-free graphs but is polynomial-time solvable on P_4-free graphs. Notice that the naive brute-force algorithm that tries all 2-partitions of non-terminal vertices runs in time $O(2^k n^{O(1)})$, where $k = n - |Z_1 \cup Z_2|$. This shows that 2-Disjoint Connected Subgraphs is fixed-parameter tractable when parameterizing by the number of non-terminals. However, Cygan et al [1] show that breaking this $O^*(2^k)$ barrier for the number k of non-terminals would contradict the Strong Exponential Time Hypothesis, and that a polynomial kernel for this parameterization would imply $NP \subseteq coNP/poly$. Paulusma and van Rooij [9] gave an algorithm with runtime $O^*(1.2051^n)$ for P_6-free graphs and asked whether it was possible to solve the problem in general graphs faster than $O(2^n n^{O(1)})$. This question was recently answered affirmatively by Cygan et al [1] who gave an algorithm for 2-Disjoint Connected Subgraphs on general graphs, based on the branch and reduce technique, with runtime $O^*(1.933^n)$.

Our algorithm for 2-Disjoint Connected Subgraphs on general graphs will be based on Enumeration of Minimal T-Connecting Sets and have runtime $O^*(1.7804^n)$.

Our paper is organized as follows. In Section 2 we give the mian definitions. In Section 3 we address the enumeration of induced paths starting in v and ending in a vertex from a given set R, with no intermediate vertices in R. In Section 4 we give an algorithm for Enumeration of Minimal T-Connecting Sets. In Section 5 we apply this enumeration algorithm to solve the 2-Disjoint Connected Subgraphs problem. We end in Section 6 with some questions and relating our main enumeration task to the minimum steiner tree problem with unit weights.

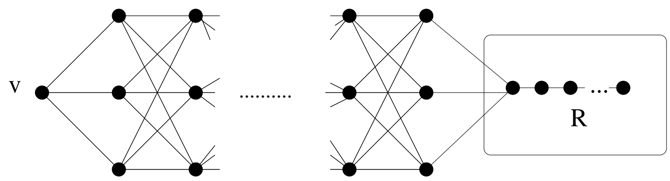

Figure 1 The number of induced paths between vertex v and a vertex of $N(R)$ (the rightmost column of 3 vertices) is $3^{\frac{n-|R|-1}{3}}$. Since each such path P has branch depth $b(P) = n - |R| - 1$ this graph shows tightness of Lemma 2. R induces a connected graph so the number of minimal $R \cup \{v\}$-connecting sets is also $3^{\frac{n-|R|-1}{3}}$.

2 Definitions

We deal with simple undirected graphs and use standard terminology. For a graph $G = (V, E)$ and $S \subseteq V$ we denote by $G[S]$ the graph induced by S. An induced subgraph $G[S]$ for $S \subset V$ is called connected if any pair of vertices of S are connected by a path in $G[S]$. We may also denote the vertex set of a graph G by $V(G)$. We denote by $N[S]$ the set of vertices that are in S or have a neighbor in S, and let $N(S) = N[S] \setminus S$.

A path P of a graph G is a sequence of vertices $(v_1, v_2, \ldots, v_q)$ such that $v_j v_{j+1} \in E$ for $1 \leq j < q$, and the path is called induced if $G[\{v_1, v_2, \ldots, v_q\}]$ has no other edges. A subpath of P is of the form $(v_1, v_2, \ldots, v_i)$ for some $i \leq q$.

Contracting an edge uv into vertex v in a graph G is defined as the operation of adding, for every vertex $w \in N(u) \setminus N[v]$, the edge vw to G if it is not already present, and then deleting u and all edges incident to u. Notice that a graph is connected after the contraction operation if and only if it was connected before the contraction operation.

For a subset of vertices T a superset S of T is T-connecting if $G[S]$ is connected, and S is minimal T-connecting if no strict subset of S is T-connecting.

Given a graph $G = (V, E)$, a vertex set $T \subset V$, a vertex $v_1 \in V \setminus T$, and an induced path $P = (v_1, v_2, ..., v_q)$ in $G[V \setminus T]$, we define the *branch depth* of path P to be

$$b(P) = |N[\{v_1, v_2, ..., v_{q-1}\}]| - 1.$$

3 Induced paths from a vertex to a set of vertices

It is well-known that the set of induced paths between a pair of vertices in an n-vertex graph can be enumerated in $O^*(3^{\frac{n-2}{3}})$ time. We have not been able to find a written proof of this in the literature. In the following theorem the induced paths between a pair of vertices is a special case, thus providing a generalization of a well known result.

▶ **Theorem 1.** *Given a graph $G = (V, E)$, a vertex $v \in V$ and $R \subseteq V \setminus N[v]$, we can enumerate all induced paths from v to a vertex of $N(R)$, with no intermediate vertex in $N[R]$, in time $O^*(3^{\frac{|V \setminus R|-1}{3}})$.*

We actually want the paths from v to R, but since these paths must have the second-to-last vertex in $N(R)$ we state the result as above. Theorem 1 will follow from Lemma 2, which is stated in terms of branch depth of paths in order to be used for the branching algorithm in the next section. Since the branch depth of each induced path from v to $N(R)$, with no intermediate vertex in $N(R)$, is at most $|V \setminus R| - 1$, Theorem 1 will follow from Lemma 2 below and is tight up to polynomial factors, see Figure 1.

▶ **Lemma 2.** *Given a graph $G = (V, E)$, a vertex $v_1 \in V$, $R \subseteq V \setminus N[v_1]$, and an integer t. Then there exists at most $3^{\frac{t}{3}}$ induced paths $P = (v_1, v_2, ..., v_q)$ in G such that*

- $b(P) \leq t$,
- $v_i \notin N[R]$ *for* $1 \leq i \leq q - 1$, *and*
- $v_q \in N(R)$.

Furthermore all these paths can be enumerated in $O^*(3^{\frac{t}{3}})$ time.

Proof. Our objective is to enumerate all induced paths starting in v_1 and ending in a vertex of $N(R)$ with none of the intermediate vertices of the path in $N[R]$. Let us call such paths full paths. Note that vertices of R will never be used in a full path and we can stop the search at the moment a vertex of $N(R)$ is reached.

The proof is by induction on the size of integer t which is an upper bound for the branch depth of all involved paths. We will strengthen the induction hypothesis. Let $\mathcal{S}_t$ be the set of paths of branch depth at most t that are sub-paths of full paths. Let $\mathcal{P}_t$ be the inclusion maximal paths in $\mathcal{S}_t$. For example, if $P = (v_1, v_2, ..., v_q)$ is a full path and $P' = (v_1, v_2, ..., v_j)$ a sub-path of P of branch-depth $b(P') = t$ then $P' \in \mathcal{P}_t$ but the sub-path $P'' = (v_1, v_2, ..., v_{j'})$ for $j' < j$ is not inclusion maximal in $\mathcal{S}_t$ so $P'' \notin \mathcal{P}_t$. Our induction hypothesis will be that $|\mathcal{P}_t| \leq 3^{\frac{t}{3}}$. This is a strengthening as the Lemma states this bound only for the number of full paths.

Since $R \subseteq V \setminus N[v_1]$ each full path contains at least two vertices v_1 and v_2 for some $v_2 \in N(v_1)$ and thus each full path has branch depth at least $|N[v_1]| - 1$. We use $t_{base} = |N[v_1]| - 1$ as the base case of the induction proof. There are at most t_{base} different inclusion maximal sub-paths of full paths starting in v_1 having branch depth at most t_{base} and these sub-paths are of the form (v_1, v_2) for a vertex $v_2 \in N(v_1)$. As the branch depth of all these paths is exactly t_{base} and no full path has smaller branch depth we thus have $|\mathcal{P}_{t_{base}}| \leq t_{base}$ and establish the base case by proving that $r \leq 3^{r/3}$ for $r \geq 1$. The argument is by induction on r. Base cases are $r \in \{0, 1, 2, 3\}$ which holds as $3^{r/3}$ for these values are lower bounded by $\{1, 1.44, 2.08, 3\}$, respectively. Now by the induction hypothesis assume that $r - 1 \leq 3^{(r-1)/3}$ and let us argue that $r \leq 3^{r/3}$ which can be rewritten as $r - 1 \leq 3^{(r-1)/3}(3^{1/3} - 1/(3^{(r-1)/3}))$. Given this it suffices to argue that $1 \leq (3^{1/3} - 1/(3^{(r-1)/3})$ for $r > 3$. As $3^{1/3} > 1.44$ and $1/(3^{(r-1)/3}) \leq 0.34$ for $r \geq 4$ the base case holds.

Now back to the induction step of the main proof. Our objective is to argue that $|\mathcal{P}_t| \leq 3^{t/3}$ under the assumption that $|\mathcal{P}_j| \leq 3^{j/3}$ for $1 \leq j < t$. Notice that for all pairs of paths P and P' where $P = (v_1, v_2, ..., v_i)$ and $P' = (v_1, v_2, ...v_i, v_{i+1})$ we have that $b(P) < b(P')$ as $v_{i+1} \in N[v_i]$ but $v_{i+1} \notin N[v_{i-1}]$ so $b(P) = |N[\{v_1, v_2, ..., v_{i-1}\}]| - 1 < |N[\{v_1, v_2, ..., v_i\}]| - 1 = b(P')$. Note also that there are exactly $b(P') - b(P)$ different paths that can be obtained by adding a single vertex v_{i+1} to P and all of these are contained in $\mathcal{P}_{b(P')}$. Notice also that $P \notin \mathcal{P}_{b(P')}$.

The set of paths $\mathcal{P}_j$ can be obtained from the set $\mathcal{P}_{j-1}$ as follows: Initially let $\mathcal{P}_j = \emptyset$. For every path $P = (v_1, v_2, ..., v_i) \in \mathcal{P}_{j-1}$, if $j = b(P) + |N[v_i] \setminus N[\{v_1, v_2, ..., v_{i-1}\}]|$ then add path $P = (v_1, v_2, ..., v_{i+1})$ to $\mathcal{P}_j$ for every vertex $v_{i+1} \in N[v_i] \setminus N[\{v_1, v_2, ..., v_{i-1}\}]$, else add path P to $\mathcal{P}_j$. This procedure is correct as $P \in \mathcal{P}_{j-1}$ means $b(P) \leq j - 1$ and the branch depth always increases when a new vertex is added, the only paths of $\mathcal{P}_j \setminus \mathcal{P}_{j-1}$ are the paths of branch depth exactly j and finally no sub-path of a path is allowed as the set is inclusion maximal.

Let $\mathcal{P}_{j-1}^{j-k}$ be the set of paths in $\mathcal{P}_{j-1}$ of branch depth exactly $j - k$. As there is exactly $k = |N[v_i] \setminus N[\{v_1, v_2, ..., v_{i-1}\}]|$ ways to extend the path ending in vertex v_{i-1} by a single vertex v_i to a path of branch depth j and as $r \leq 3^{r/3}$ for every integer $r > 0$ we can make the following sum:

$$|\mathcal{P}_t| \leq \sum_{k=1}^{t-1} |\mathcal{P}_{t-1}^{t-k}| \cdot 3^{k/3}$$

where our goal is to show that this adds up to at most $3^{t/3}$. This is done by refining the expression recursively, let $r \in [1, t-1]$ be the largest integer such that $|\mathcal{P}_{t-1}^r| > 0$. For every path $P' = (v_1, \ldots, v_{p+1}) \in \mathcal{P}_{t-1}^r$ we repeat the following procedure: For the subpath $P'' = (v_1, \ldots, v_p)$ we can notice that there are exactly $a = |N(v_p) \setminus N[\{v_1, \ldots, v_{p-1}\}]|$ ways to extend P'' to a path $P''' \in \mathcal{P}_{t-1}^r$ we can remove these a paths from $\mathcal{P}_{t-1}^r$ and add P'' to $\mathcal{P}_{t-1}^{r-a}$ without reducing the sum on the right-hand side as $a \leq 3^{a/3}$. Repeat this recursively until $r = |N(v_1)|$ is the highest number such that $|\mathcal{P}_{t-1}^r| > 0$. As all paths we are searching for have branch depth at least $|N(v_1)|$ it follows that $\mathcal{P}_{t-1}^r$ for $r = |N(v_1)|$ is the only non-empty set. Thus we get that

$$|\mathcal{P}_t| = |\mathcal{P}_{t-1}^r| \cdot 3^{(t-r)/3} \leq 3^{r/3} \cdot 3^{(t-r)/3}$$

and the proof is completed.

By turning the induction proof into an algorithm it is not hard to see that all paths of branch depth at most t can be enumerated in $O^*(3^{t/3})$ time. ◀

The resulting enumeration algorithm is optimal to within polynomial factors, see Figure 1.

4 Enumeration of Minimal T-Connecting Sets

Theorem 1 with $|R| = 1$ can be viewed as an enumeration of all minimal T-connecting sets when $T = \{u, v\}$. We now generalize this approach to an arbitrary terminal set T by a branching algorithm. The following observation will be used to simplify our branching algorithm.

▶ **Lemma 3.** *Given $G = (V, E)$, $T \subseteq V$, and two vertices $u, v \in T$ such that $uv \in E$. Let G' be the graph obtained by contracting edge uv into v. Then there is a one to one mapping between Minimal T-Connecting Sets in G and Minimal $T \setminus \{u\}$-Connecting Sets in G'.*

Proof. For every Minimal T-Connecting Set S in G we can contract edge uv and obtain a minimal $T \setminus \{u\}$-Connecting Sets $S' = S \setminus \{u\}$ in G'. For every minimal $T \setminus \{u\}$-Connecting Sets S' in G' we can observe that $G[S' \cup \{u\}]$ is a T-Connecting Set in G and it is also minimal as $u \in T$. ◀

Consider **Algorithm Main Enumeration**. It will solve Enumeration of Minimal T-Connecting Sets for any graph $G = (V, E)$ and $T \subseteq V$. Let us first give the informal intuition for the algorithm. We fix a vertex $u \in T$ and using the algorithm of Lemma 2 we find all induced paths from u to $N(T \setminus \{u\})$. For each of these paths P we again call the algorithm of Lemma 2, but now on the graph G' where the path P together with the vertices of T that P has in its neighborhood, is contracted into u. The path we find in G' will in G start at an arbitrary vertex of P or a neighbor in T and we see that we start forming a tree of paths. We carry on recursively in this way until the collection of paths spans all of T, note however that the vertices of these paths may induce a graph containing cycles. To avoid repeating work we label vertices by a total order and use this ordering to guide the recursive calls.

Algorithm Main Enumeration
Input: A graph $G = (V, E)$ and terminal set $T \subseteq V$
Output: A family of sets containing all Minimal T-Connecting Sets
begin
  assign each vertex a unique *label* between 1 and $|V|$
  choose $u \in T$
  MCS$(\emptyset, \emptyset)$
end

Procedure MCS(C, X)
Parameter C: vertex set used to connect T
Parameter X: vertices of $N(C_u)$ not to explore in this call
begin
if $G[T \cup C]$ is connected then **output** $T \cup C$
else
  set $C_u \supseteq C$ as vertex set of connected component of $G[T \cup C]$ containing u
  set $T' = T \setminus C_u$ i.e. the terminals not yet connected to u by C
  set G' to be graph obtained from G by contracting edges of $G[C_u]$ to u
  call the algorithm of Lemma 2 on $G'[V(G') \setminus X]$ with $v_1 = u$ and $R = T'$
  for every path $P = (v_1, v_2, \ldots, v_q)$ output by that call
    MCS$(C \cup \{v_2, \ldots, v_q\},\ X \cup \{w \in N(C_u) : label(w) < label(v_2)\})$
  end-for
end

▶ **Lemma 4.** *Given $G = (V, E)$, $T \subseteq V$ and $|T| \leq n/3$ Algorithm Main Enumeration will:*
1. *output every Minimal T-Connecting Set of G,*
2. *output, for any integer $r \in [0..|V \setminus T|]$, at most $\binom{|V \setminus T|}{|T|-2} \cdot 3^{r/3}$ vertex sets $S \supseteq T$ such that $|N[S] \setminus T| \leq r$, and*
3. *run in $O^*(\binom{|V \setminus T|}{|T|-2} \cdot 3^{|V \setminus T|/3})$ time.*

Proof. 1.) Let us first argue that every Minimal T-connecting vertex set is output by the algorithm. In the case where $|T| \leq 1$ the single vertex set T is output by the algorithm. In the remaining cases $|T| > 1$.

Let S be a minimal T-connecting vertex set. Our goal will be to show that there will be a call MCS(C, X) performed by the algorithm in which $T \cup C = S$. Initially $C = X = \emptyset$ so we trivially have $T \cup C \subseteq S$ and $X \cap S = \emptyset$. Consider the call MCS(C, X) where we have $|T \cup C|$ maximized under the constraint $T \cup C \subseteq S$ and $S \cap X = \emptyset$. We show by contradiction that $T \cup C = S$ for this call MCS(C, X). Assume, by sake of contradiction, that there is a terminal vertex not in C_u, i.e. not in the component of $G[T \cup C]$ containing u, i.e. that $T' \neq \emptyset$. Let v_2 be the lowest numbered vertex of $(N(C_u) \cap S) \setminus X$. As S is minimal we have that $G[S]$ is connected but $G[S \setminus \{v_2\}]$ is not connected. By the minimality of S we have that $G'[S']$ is connected but $G'[S' \setminus \{v_2\}]$ is not connected for $S' = (S \setminus C_u) \cup \{u\}$. Vertex v_2 is not a vertex of $C \cup T$ and as S is minimal we have that each connected component of $G'[S' \setminus \{v_2\}]$ contains a vertex of T'. If this was not the case, this component could simply be removed from S without changing the connectivity between vertices of T. Let B be a connected component of $G'[S' \setminus \{v_2\}]$ not containing u. By the previous arguments B contains a vertex of T'. Therefore the call of the algorithm in Lemma 2 on graph G' with $R = T'$ will find a path $P = (u, v_2, \ldots, v_q)$ with all vertices in S and with v_q a neighbor of a vertex of T' in B and containing only vertex v_2 from $N(C_u)$. This would lead to a recursive call where C would be updated to $C \cup \{v_2, \ldots, v_q\} \subseteq S$, and to X there would not be added any vertices of S as v_2 had lowest label among all vertices in $(N(C_u) \cap S) \setminus X$, contradicting the maximality of $|T \cup C|$ under the constraint $T \cup C \subseteq S$ and $S \cap X = \emptyset$.

2.) We bound the number of recursive calls in the algorithm and thus also the number of vertex sets that is output. Our objective will be to prove that the number of recursive calls $\mathrm{MCS}(C, X)$ where $x = |X|$, $r = |N[C_u] \setminus T|$, and p is the number of times a path is added to C (which is equal to the depth of the recursion) is at most $\binom{x+p}{p-1} \cdot 3^{r/3}$. As $X \subset N(C_u)$ by the construction of the algorithm, $p \leq |T| - 1$, and at least one vertex is added to C for each found path so $p \leq |C|$, we have that $x + p \leq |N[C_u] \setminus T| = r$. Given that $|T| \leq n/3$ and thus $|V \setminus T| \geq 2|T|$ it is clear that $\binom{|V \setminus T|}{|T|-2} \geq \binom{x+t}{t-1}$ and the claim of the lemma follows.

The proof will be by induction on $\ell = x + p$. We assume without loss of generality that $|T| \geq 2$. The first call is $\mathrm{MCS}(\emptyset, \emptyset)$ in which case $p = 0$, and this is in fact the only call where $x + p \leq 0$. The execution of $\mathrm{MCS}(\emptyset, \emptyset)$ will call the algorithm of Lemma 2 on G' with $v_1 = u$ and $R = T'$ and make a recursive call $\mathrm{MCS}(C, X)$ for each path P output by the algorithm of Lemma 2. Consider such a call $\mathrm{MCS}(C, X)$ originating from path P. This call will have $x = |X| \geq 0$, $p = 1$, and it will have $r = |N[C_u] \setminus T| \geq b(P)$. The number of paths P with $b(P) \leq r$ output by the algorithm of Lemma 2 applied to the execution of $\mathrm{MCS}(\emptyset, \emptyset)$ on G' with $v_1 = u$ and $R = T'$ is at most $3^{r/3}$. Since $3^{r/3} \leq \binom{x+p}{p-1} \cdot 3^{r/3}$ for $p = 1$ we have just established the base case $\ell = x + p \leq 1$ this also covers all cases where $p \leq 1$ in our induction.

In the induction step we consider the case where $\ell = x + p \geq 2$ and $p > 1$. Let $\mathrm{MCS}(C', X')$ be a call and let $x' = |X'|$, $r' = |N[C'_u] \setminus T|$, and p' be the number of paths added, or equivalently the depth of the recursion. By the induction hypothesis we assume that the bound holds for the number of calls $\mathrm{MCS}(C', X')$ where $x' + p' \leq x + p - 1$.

Every call $\mathrm{MCS}(C, X)$ where $x + p = \ell$ is created by a call $\mathrm{MCS}(C', X')$ and a path $P = (v_1, v_2, \ldots, v_q)$ such that $C = C' \cup \{v_2, \ldots, v_q\}$, $p' = p - 1$, and $X = X' \cup \{w \in N(C'_u) : \mathit{label}(w) < \mathit{label}(v_2)\}$. As each vertex from $N(C'_u) \setminus X'$ chosen as v_2 will create a unique size on the set $X = X' \cup \{w \in N(C'_u) : \mathit{label}(w) < \mathit{label}(v_2)\}$ for the next recursive call there is at most one choice for v_2 starting from a fixed $\mathrm{MCS}(C', X')$ when it should lead to a recursive call $\mathrm{MCS}(C, X)$ where $x + p = \ell$. However, there are choices for the sub-path vertices $(v_3, \ldots, v_q)$, but these vertices can be chosen only among $V \setminus (N[C'_u] \cup T)$, since v_2 is fixed in $N(C'_u)$ and the path P is induced. Note that any such sub-path has branch-depth at most $|N[C_u] \setminus (N[C'_u] \cup T)|$. We can use Lemma 2 to bound the number of such sub-paths, as follows. By applying Lemma 2 to the graph we get from $G[V \setminus (N(C'_u) \setminus \{v_2\})]$ by contracting $C'_u \cup \{v_2\}$ to $u = v_1$ and with $R = T \setminus C'_u$ we deduce that the number of such sub-paths is at most $3^{(|N[C_u] \setminus (N[C'_u] \cup T)|)/3}$.

This means that the number of calls $\mathrm{MCS}(C, X)$ where $x + p = \ell$ is at most the number of calls $\mathrm{MCS}(C', X')$ where $C' \subseteq C$, $X' \subseteq X$ thus $x' \leq x$, and $p' = p - 1$, times $3^{(|N[C_u] \setminus (N[C'_u] \cup T)|)/3}$. By the induction hypothesis we have that the number of calls $\mathrm{MCS}(C', X')$ where $x' + p' < \ell$ is at most $\binom{x+p-1}{p-1-1} \cdot 3^{|N[C'_u] \setminus T|/3}$. Multiplying these two factors we get $\binom{x+(p-1)}{(p-1)-1} \cdot 3^{|N[C'_u] \setminus T|/3} \cdot 3^{(|N[C_u] \setminus (N[C'_u] \cup T)|)/3}$ which can be simplified to $\binom{x+(p-1)}{(p-1)-1} \cdot 3^{(|N[C_u] \setminus T|)/3}$.

Thus it remains to bound the number of calls $\mathrm{MCS}(C', X')$ that can make a new recursive call $\mathrm{MCS}(C, X)$ where $x + p = \ell$ to be at most $\binom{x+p}{p-1}$. We know that each call $\mathrm{MCS}(C', X')$ can only make calls where $x + p = \ell$ when it uses the unique vertex $v_2 \in N(C'_u) \setminus X$ as the second vertex of the path. Thus it suffices to count these calls, and let y be the number of such calls. We have that

$$y \leq \sum_{i=0}^{x} \binom{i+p-1}{p-1-1} \cdot 3^{(|N[C_u]\setminus T|)/3}$$

Using the standard observation that $\sum_{k=0}^{n} \binom{k}{m} = \binom{n+1}{m+1}$ we can conclude that $y \leq \binom{x+p}{p-1} \cdot 3^{(|N[C_u]\setminus T|)/3}$ and the proof is completed.

3.) In the previous claim we bounded the number of recursive calls in the algorithm to $\binom{|V\setminus T|}{|T|-2} \cdot 3^{r/3}$ vertex sets $S \supseteq T$ such that $|N[S] \setminus T| \leq r$ and $|T| \leq n/3$, and as Lemma 2 ensures that all paths in a single call can be enumerated within a polynomial delay it follows that the polynomial bound holds. ◀

Using Lemma 4 we can make the following conclusion.

▶ **Theorem 5.** *For an n vertex graph $G = (V, E)$ and a terminal set $T \subseteq V$ where $|T| \leq n/3$ there is at most $\binom{n-|T|}{|T|-2} \cdot 3^{(n-|T|)/3}$ minimal T-connecting vertex sets and these can be enumerated in $O^*(\binom{n-|T|}{|T|-2} \cdot 3^{(n-|T|)/3})$ time.*

5 The 2-Disjoint Connected Subgraphs problem

Let us now use Theorem 5 to solve the 2-Disjoint Connected Subgraphs problem. Recall that the problem is defined as follows:

2-Disjoint Connected Subgraphs
Input: A connected graph $G = (V, E)$ and two disjoint subsets of vertices $Z_1, Z_2 \subseteq V$.
Question: Does there exist two disjoint subsets A_1, A_2 of V, with $Z_1 \subseteq A_1, Z_2 \subseteq A_2$ and $G[A_1]$, $G[A_2]$ both connected?

▶ **Theorem 6.** *There exists a polynomial space algorithm that solves the* 2-Disjoint Connected Subgraphs *problem in $O^*(1.7804^n)$ time.*

Proof. Let us assume without loss of generality that $|Z_1| \leq |Z_2|$ and let $0 < \alpha \leq 0.5$ be a constant such that $\alpha n = |Z_1|$. The algorithm has a first stage that finds a list of potential candidates for A_1 and a second stage that checks each candidate to see if it can be used as a solution. In the first stage we choose between two different strategies depending on whether or not $\alpha > 0.0839$.

Consider first the case where $\alpha \leq 0.0839$. Vertices of Z_2 are of no use when searching for a potential set A_1 so it suffices to consider the graph $G[V \setminus Z_2]$. By the algorithm for Enumeration of Minimal T-Connecting Sets of Theorem 5 we know that for $|Z_1| \leq n/3$ in the graph $G[V \setminus Z_2]$ all minimal Z_1-connecting sets can be enumerated in $O^*(\binom{n-|Z_1|-|Z_2|}{|Z_1|-2} \cdot 3^{(n-|Z_1|-|Z_2|)/3})$ time. As $|Z_1| \leq |Z_2|$ it is clear that $\alpha n \leq |Z_2|$. The number $|Z_2|$ only contributes negatively so we can observe that

$$\binom{n-|Z_1|-|Z_2|}{|Z_1|-2} \cdot 3^{(n-|Z_1|-|Z_2|)/3} \leq \binom{(1-2\alpha)n}{\alpha n - 2} \cdot 3^{(1-2\alpha)n/3}.$$

By using the Stirling approximation we know that $\binom{(1-2\alpha)n}{\alpha n-2}$ is $O^*((\frac{\beta^\beta}{\alpha^\alpha \cdot (\beta-\alpha)^{(\beta-\alpha)}})^n)$ where $\beta = (1-2\alpha)$. It is not hard to verify that the maximum value of $\binom{(1-2\alpha)n}{\alpha n-2} \cdot 3^{(1-2\alpha)n/3}$ for $0 < \alpha \leq 0.0839$ occurs when $\alpha = 0.0839$ and that $\binom{(1-2\alpha)n}{\alpha n-2} \cdot 3^{(1-2\alpha)n/3} \leq 1.7804^n$ for

$\alpha = 0.0839$. Thus, we can conclude that when $\alpha \leq 0.0839$ a list of all minimal Z_1-connecting sets can be found in time $O^*(1.7804^n)$.

Consider now the case where $\alpha > 0.0839$. In this case the algorithm simply loops over all subsets of $V \setminus (Z_1 \cup Z_2)$ to list every vertex subset $A \subseteq (V \setminus Z_2)$ where $Z_1 \subseteq A$. As $\alpha > 0.078$ and $\alpha n = |Z_1| \leq |Z_2|$ we get that the number of such subsets is at most $2^{n-2\alpha} \leq 1.7804^n$ and they can be found in $O^*(1.7804^n)$ time.

For the second stage of the algorithm, for every listed set A, the algorithm tests if vertices of Z_2 are contained in the same connected component of $G \setminus A$ and if so the algorithm returns the solution with $A_1 = A$ and A_2 being the vertices of the connected component of $G \setminus A$ containing Z_2. This is clearly a solution to the problem. Conversely, if there is a solution A_1, A_2 to the problem, then there is clearly one where A_1 is a minimal Z_1-connecting set.

Finally, observe that the algorithm uses polynomial space as a simple branching algorithm is used for both cases. ◀

Conclusion

The graph in Figure 1 shows that our algorithm for Enumeration of Minimal T-Connecting Sets given by Theorem 5 is optimal, up to polynomial factors, for the case $|T| = 2$. Is the algorithm optimal, up to polynomial factors, also for larger T, let us say $|T| \leq 0.1n$? If we simply run our algorithm for Enumeration of Minimal T-Connecting Sets and then for an output set S with smallest cardinality return $S \setminus T$ we solve the following problem

Minimum Connecting Set
Input: Graph G, a set of terminal vertices T
Output: A minimum-size $W \subseteq V$ with $G[T \cup W]$ connected

This problem is closely related to

Steiner Tree with unit weights
Input: a graph G and a set of terminal vertices T
Output: A minimum-size $F \subseteq E$ such that $G[F]$ is a connected graph containing T

Since the minimum number of edges needed to connect a set of Steiner points will always induce a tree the following is clear

▶ Observation 7. Given $G = (V, E), T \subseteq V$, let W and F be solutions to Minimum Connecting Set and Steiner Tree with unit weights respectively. We then have $|F| = |W| + |T| - 1$.

Using our algorithm for Enumeration of Minimal T-Connecting Sets we solve both these problems in time $O^*(\binom{|V \setminus T|}{|T|-2} \cdot 3^{\frac{|V \setminus T|}{3}})$ which is upper bounded by $O^*(1.8778^n)$ when balanced with the standard brute force search.

The best exact algorithm for Steiner Tree with unit weights is by Nederlof [8] and has runtime $O^*(1.3533^n)$ using polynomial space. Parameterizing by the number of terminals T the same paper gives an FPT algorithm for Steiner Tree with unit weights with runtime $O^*(2^{|T|})$.

References

1 Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Solving the 2-disjoint connected subgraphs problem faster than 2 n. In David Fernández-Baca, editor, *LATIN*, volume 7256 of *Lecture Notes in Computer Science*, pages 195–206. Springer, 2012.

2 Fedor V. Fomin, Serge Gaspers, Artem V. Pyatkin, and Igor Razgon. On the minimum feedback vertex set problem: Exact and enumeration algorithms. *Algorithmica*, 52(2):293–307, 2008.

3 Fedor V. Fomin, Fabrizio Grandoni, Artem V. Pyatkin, and Alexey A. Stepanov. Combinatorial bounds via measure and conquer: Bounding minimal dominating sets and applications. *ACM Transactions on Algorithms*, 5(1), 2008.

4 Fedor V. Fomin and Yngve Villanger. Treewidth computation and extremal combinatorics. *Combinatorica*, 32(3):289–308, 2012.

5 Chris Gray, Frank Kammer, Maarten Löffler, and Rodrigo I. Silveira. Removing local extrema from imprecise terrains. *Comput. Geom.*, 45(7):334–349, 2012.

6 David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. On generating all maximal independent sets. *Inf. Process. Lett.*, 27(3):119–123, 1988.

7 J. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:23–28, 1965. 10.1007/BF02760024.

8 Jesper Nederlof. Fast polynomial-space algorithms using möbius inversion: Improving on steiner tree and related problems. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *ICALP (1)*, volume 5555 of *Lecture Notes in Computer Science*, pages 713–725. Springer, 2009.

9 Daniël Paulusma and Johan M. M. van Rooij. On partitioning a graph into two connected subgraphs. *Theor. Comput. Sci.*, 412(48):6761–6769, 2011.

10 Neil Robertson and Paul D. Seymour. Graph minors .xiii. the disjoint paths problem. *J. Comb. Theory, Ser. B*, 63(1):65–110, 1995.

11 Pim van 't Hof, Daniël Paulusma, and Gerhard J. Woeginger. Partitioning graphs into connected parts. *Theor. Comput. Sci.*, 410(47-49):4834–4843, 2009.