A Boundary Class for the k-Path Partition Problem

Nicholas Korpelainen [1]

Department of Electronics, Computing & Mathematics
University of Derby
Derby, UK

Abstract

We establish the first known *boundary class* for the k-path partition problem and deduce that for a graph class defined by finitely many minimal forbidden induced subgraphs, the k-path partition problem remains NP-hard unless one of the forbidden induced subgraphs is a subcubic tree (a tree of maximum degree at most 3) with at most one vertex of degree 3.

Keywords: hereditary graph classes, boundary properties

1 Introduction to the k-path partition problem

The k-path partition problem (k-PP) is, given a graph G, the problem of finding a minimum number of vertex-disjoint (not necessarily induced) paths of length at most k that partition $V(G)$.

The k-path partition problem has several real-life applications, for instance in the field of broadcasting in computer and communication networks [7]. The problem is known to be NP-complete in the class of all graphs [3]. To

[1] Email: N.Korpelainen@derby.ac.uk

get an intuition for possible applications, one might consider the problem of minimising the number of postal delivery vans needed to service a city, where each van can only service a limited amount of customers (or can only drive a limited distance) on its daily route, visiting each customer at most once. Clearly this problem can be made to correspond to minimising the number of vans needed to service all customers.

Let us also introduce a useful variant of this problem, called the P_k-partition problem: The P_k-partition problem is, given a graph G, the decision problem of deciding whether $V(G)$ can be partitioned into vertex-disjoint subgraphs isomorphic to P_k .

Each of the above two algorithmic problems has an 'induced variant', (i.e. the induced k-path partition problem and the induced P_k-partition problem), each defined by the additional requirement that the paths in the partitions must be induced subgraphs of the underlying graph.

In order to highlight the usefulness of the P_k-partition problem, we note that whenever this problem is NP-hard on a graph class X, then the k-path partition problem must also be NP-hard on X. A similar statement obviously holds for the induced variants of the two problems, respectively.

An overview of the complexity status of the k-path partition problem for various graph classes is given in Figure 1. It is of particular note that although the problem is known to be NP-complete on the class of convex graphs [2] (a superclass of biconvex graphs), and polynomial-time solvable for bipartite permutation graphs [5] (a subclass of biconvex graphs), the complexity status remains an open problem for the class of biconvex graphs.

2 A boundary class

In a paper by Steiner, the author used a reduction from EXACT COVER BY 3-SETS to show that the 3-path partition problem is NP-complete on comparability graphs [5]. Later, similar ideas were used in [4], with a reduction from k-DM (the k-dimensional matching problem), to prove that the P_k-partition problem (and the induced P_k-partition problem) remains NP-complete on bipartite graphs of maximum degree 3, for any fixed $k \geq 3$. As discussed in the previous section, this is enough to show NP-completeness of the k-path partition problem for the same graph class. In this section we will extend the latter proof with the aim of discovering the first boundary class for the k-path partition problem (k-PP).

A graph class X will be called k-PP-*easy* if the k-path partition problem is polynomial-time solvable for graphs in X, and k-PP-*tough* otherwise. If

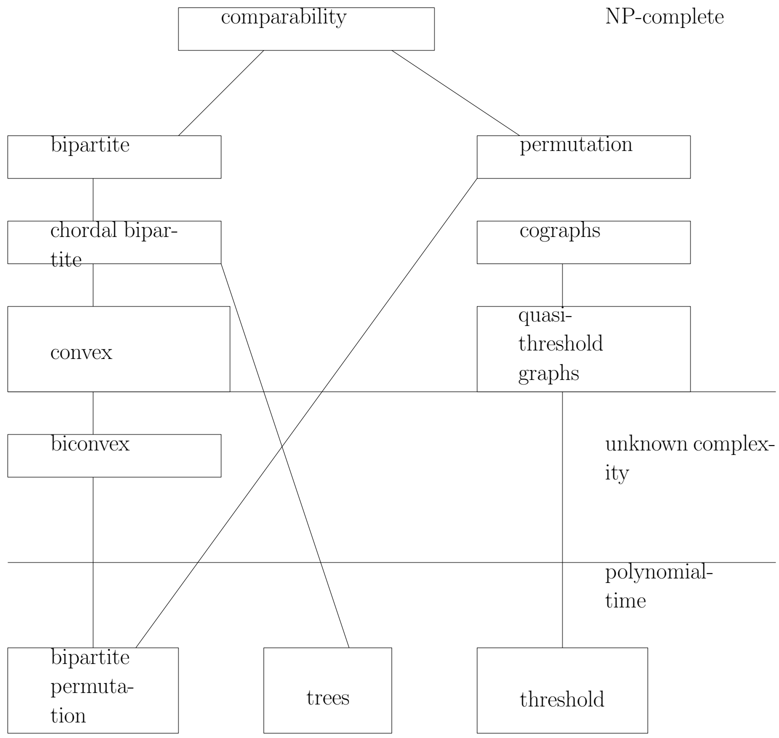

Fig. 1. The computational complexity of the k-path partition problem

$P \neq NP$, the family of k-PP-tough classes is disjoint from that of k-PP-easy classes, in which case the problem of characterisation of these two families arises. We want to characterise the family of k-PP-easy classes in terms of minimal classes that do not belong to this family. Unfortunately, a k-PP-tough class may contain infinitely many k-PP-tough subclasses, which makes the task of finding minimal k-PP-tough classes impossible. To overcome this difficulty, we employ the notion of a boundary class, originally introduced by Alekseev in the context of the independent set problem [1]. The notion can be defined (in this context) as follows.

A class of graphs $\mathcal{S}$ will be called a *limit class* for the k-path partition problem if $\mathcal{S} = \bigcap_{i=1}^{\infty} \mathcal{S}_i$, where $\mathcal{S}_1 \supseteq \mathcal{S}_2 \supseteq \ldots$ is a sequence of k-PP-tough classes. A minimal limit class will be a *boundary* class for the problem in question.

The importance of the notion is due to two facts: (1) every k-PP-tough class contains a boundary class; (2) every graph class defined by finitely many minimal forbidden induced subgraphs is k-PP-tough if and only if it contains a boundary class (again by adapting ideas from [1]).

We define H_i and $S_{i,j,k}$ as the graphs represented in Figure 2.

Definition 2.1 We define S_i to be the class of $(C_3, C_4, \dots, C_i, H_1, H_2, \dots, H_i)$-free bipartite graphs of maximum degree 3.

Lemma 2.2 *Let G be a graph and e an edge in G. If G' is the graph obtained from G by subdividing the edge e exactly by mk times, for some positive integers k and m, then G has a P_k-partition if and only if G' has a P_k-partition.*

Proof. Denote the endpoints of e by a and b. In G', we denote the subdivided e by $S := (a, s_1, s_2, \dots, s_{mk}, b)$.

First suppose that G has a P_k-partition $\mathcal{P}$. If e does not belong to any subgraph P_k in the partition, then G' has a P_k-partition $\mathcal{P}'$, which we define as the union of $\mathcal{P}$ with the m disjoint copies of P_k that cover S in G'. So we may assume that e belongs to some P_k in $\mathcal{P}$, say P.

We claim that one can construct a P_k-partition $\mathcal{P}'$ of G' by replacing P with $m+1$ disjoint copies of P_k. Suppose $P = (p_1, p_2, \dots, p_i, a, b, q_1, q_2, \dots, q_j)$, where $i + j + 2 = k$. Then we let

$$(p_1, p_2, \dots, p_i, a, s_1, s_2, \dots, s_{j+1}) \text{ and } (s_{mk-i}, s_{mk-i+1}, \dots, s_{mk}, b, q_1, q_2, \dots, q_j)$$

be two of the $m+1$ paths to replace P. It remains to find a P_k-partition of the path $(s_{j+2}, s_{j+3}, \dots, s_{mk-i-1})$, i.e. a path on $mk - (i+1) - (j+1) = (m-1)k$ vertices. There is a unique way to partition $P_{(m-1)k}$ into $m-1$ copies of P_k.

Conversely, suppose that G' has a P_k-partition $\mathcal{P}'$. If $\mathcal{P}'$ contains a P_k-partition of S, we can just delete its members from $\mathcal{P}'$ to construct a P_k-partition $\mathcal{P}$ of G. Otherwise, $\mathcal{P}'$ must contain two disjoint k-paths of the form

$$(p_1, p_2, \dots, p_i, a, s_1, s_2, \dots, s_{j+1}) \text{ and } (s_{mk-i}, s_{mk-i+1}, \dots, s_{mk}, b, q_1, q_2, \dots, q_j)$$

where $i + j + 2 = k$ (if the equation did not hold, it would be impossible to partition the rest of the vertices in S into copies of P_k, contradicting the

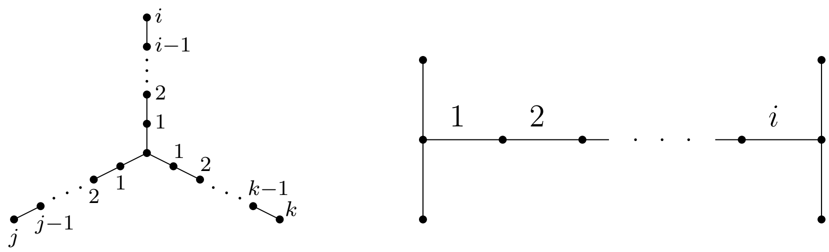

Fig. 2. Graphs $S_{i,j,k}$ (left) and H_i (right)

existence of $\mathcal{P}'$). In this case, we just delete these two paths, as well as the $m-1$ paths contained in S from $\mathcal{P}'$. Finally we add to $\mathcal{P}'$ a single path $P := (p_1, p_2, \ldots, p_i, a, b, q_1, q_2, \ldots, q_j)$. This gives us a P_k-partition $\mathcal{P}$ of G. □

Lemma 2.3 *S_i is k-PP-tough for each $i \geq 3$.*

Proof. Assuming that NP≠ P, it suffices to show that the P_k-partition problem is NP-complete on S_i. To this end, choose any positive integer m such that $mk \geq i$. Since we know that the P_k-partition problem is NP-complete on bipartite graphs of maximum degree 3, it suffices to reduce each instance of that problem to an instance of the P_k-partition problem on S_i. Given any bipartite graph G of maximum degree 3, we perform mk subdivisions on each edge of G, resulting in a new graph. Denote this new graph by G''. By repeated applications of Lemma 2.2, we know that G'' has a P_k-partition if and only if G does. Furthermore, G'' clearly belongs to S_i. This completes the reduction. □

Lemma 2.3 implies that S_i is a k-PP-tough class for any i. Therefore, $\mathcal{S} := \bigcap_{i \geq 3} S_i$ is a limit class for the k-path partition problem. It is easy to see that the graphs in $\mathcal{S}$ are precisely the graphs of maximum degree at most 3, each connected component of which is a tree with at most one cubic vertex, i.e. a graph of the form $S_{i,j,k}$ displayed in Figure 2.

Our aim is to show that $\mathcal{S}$ is a minimal limit class. To this end, we use the following Lemma, based on [1]:

Lemma 2.4 *A limit class $X = Free(M)$ for the k-path partition problem is minimal (i.e. boundary) if and only if for every element $x \in X$ there is a finite set $T \subseteq M$ such that the $Free(\{x\} \cup T)$ is k-PP-easy.*

We will apply Lemma 2.4 in the case where $X = \mathcal{S}$ and $\mathcal{A}$ is the family of k-PP-easy graph classes.

Lemma 2.5 *Let $G \in \mathcal{S}$ and suppose G has s connected components. Choose a positive integer constant t such that each connected component of G is an induced subgraph of $S_{t,t,t}$, i.e. $G \leq sS_{t,t,t}$. Then the class*

$$\mathcal{F} := Free(G, K_{1,4}, C_3, \ldots, C_{2t+1}, H_1, \ldots, H_{2t+1})$$

is k-PP-easy.

Proof. For the purposes of our proof, we may assume that $G = sS_{t,t,t}$. We claim the following:

Claim 2.6 *Let $\mathcal{T}$ be the class of graphs whose each connected component contains at most one cycle. If the k-path partition problem is polynomial-time solvable for $\mathcal{T}$, then the k-path partition problem is polynomial-time solvable for $\mathcal{F}$.*

Let us first show that the claim suffices to imply the Lemma. To do this, we prove that for any $T \in \mathcal{T}$, it is possible to find a minimum k-path partition of T in polynomial time. For this purpose, we can clearly assume that T is connected (we could otherwise consider each connected component of T in turn). If T is a tree, we can apply a result from [7] stating that the k-path partition problem is polynomial-time solvable for trees. If T contains exactly one cycle, this cycle must be an induced subgraph of T. Choose any vertex v on the cycle. For any possible k-path partition of T, its members must avoid at least one edge on the cycle which is at distance of at most $k/2$ from v. By altering which one of these $k+1$ edges is deleted, we can create $k+1$ different trees. We may assume that $k+1 \leq n := |V(T)|$ (since T certainly cannot have a path of length greater than n). Thus there are at most n different trees to check, each of which can be checked in polynomial time. Thus the claim implies the lemma.

We proceed to prove the claim, with the aim of inductively reducing each graph $F \in \mathcal{F}$ to at most $c(s) := 3^s$ graphs whose each connected component has at most one cycle. Suppose that $F \in \mathcal{F}$ has a connected component with at least two cycles. Then, by assumption, the connected component must contain two distinct induced cycles $C := C_r$ and $C' := C_l$ such that $r, l \geq 2t + 2$.

Choose a vertex w of C' that does not lie in C. Suppose v is a vertex of C that minimises $d(v, w)$, and let P' be the minimal induced path joining v and w. We claim that there exists a copy of $S_{t,t,t}$ in F, centered at v.

Clearly P' is disjoint from $C \setminus \{v\}$, by definition of v. If $d(v, w) \geq t$, then it is easy to see that F contains an induced copy of $S_{t,t,t}$, centered at v. Now assume that $d(v, w) < t$. Clearly there are two disjoint induced copies of P_t in C', each starting at w. Let us denote these two paths by P_1 and P_2. At least one of the two paths, say P_1, is disjoint from P' (otherwise F would contain an induced cycle on less than $2t + 2$ vertices, contradicting our assumption). So there exists a subpath P'' of $P' \cup P_1$, of length $t + 1$ and starting at v. Then P'' is disjoint from $C \setminus \{v\}$ (otherwise F would contain an induced cycle on less than $2t + 2$ vertices, contradicting our assumption). Also in this case, F clearly contains an induced copy of $S_{t,t,t}$, centered at v. Thus, in any case, there exists a copy of $S_{t,t,t}$ centered at v.

For any possible k-path partition of F, its members must avoid at least one neighbor of v. By altering which of these k edges is deleted, we can create 3 graphs F_2.

Now for each of the three choices of F_2, supposing that F_2 has a connected component containing at least two cycles, we can similarly find a cycle $C_2 \in F_2$ and a vertex $v_2 \in C_2$ such that there is a copy of $S_{t,t,t}$ centered at v_2. Furthermore we may assume $v_2 \neq v$, since v is of degree less than 3 in F_2, by construction. We can then proceed to create 3 graphs F_3.

Inductively, for each possible sequence $(F, F_2, \ldots, F_i)$, supposing that F_i has a connected component containing at least two cycles, we can find a cycle $C_i \in F_i$ and a vertex $v_i \in C_i$ such that there is a copy of $S_{t,t,t}$ centered at v_i. Furthermore we may assume $v_i \notin \{v_1, \ldots, v_{i-1}\}$, since the vertices of $\{v_1, \ldots, v_{i-1}\}$ are all of degree less than 3 in F_i, by construction. We can then proceed to create 3 graphs F_{i+1}.

We note that any two copies of $S_{t,t,t}$ with different central vertices in F are disjoint and without any edges between them. This follows directly from the fact that F is $(H_1, \ldots, H_{2t+1})$-free. Furthermore, since F is $(C_3, \ldots, C_{2t+1})$-free, edge deletions cannot create any new induced copies of $S_{t,t,t}$ in F; i.e. whenever F contains $S_{t,t,t}$ as a subgraph, it must contain it as an induced subgraph. In any sequence $(F, F_2, \ldots, F_{s+1})$, we have found s disjoint induced copies of $S_{t,t,t}$, such that there are no edges between any two of them, contradicting the assumption that F is G-free. Thus there are at most 3^s sequences $(F, F_2, \ldots, F_j)$, where $j \leq s$, and F_j is a graph whose each connected component contains at most one cycle.

This concludes the proof of the claim, which in turn implies the Lemma.□

Lemmata 2.4 and 2.5 together imply the following theorem.

Theorem 2.7 *$\mathcal{S}$ is a boundary class for the k-path partition problem.*

3 Concluding remarks and related open problems

We revealed the first boundary class of graphs for the k-path partition problem. Since for finitely defined graph classes (graph classes defined by forbidding finitely many minimal induced subgraphs), containment of a boundary class is a necessary and sufficient condition for k-PP-toughness, we can deduce that a finitely defined class must forbid a graph in $\mathcal{S}$ in order to be k-PP-easy. The existence of one more boundary class for this problem arises from the fact that the problem is NP-complete in the class of convex graphs (which is a subclass of chordal bipartite graphs, i.e. the class $Free(C_3, C_5, C_6, C_7 \ldots))$

[2]. This fact implies that there must exist a boundary subclass of convex graphs, i.e. a minimal class X defined by a sequence $X_1 \supseteq X_2 \supseteq X_3 \ldots$ of subclasses of convex graphs such that $X = \cap X_i$ and the problem fails to be polynomial-time solvable in each class of the sequence $X_1 \supseteq X_2 \supseteq X_3 \ldots$.

There also remain some interesting graph classes for which the complexity status of k-PP is open. The path partition problem is different from the k-path partition problem in that there is no upper bound on the lengths of the paths in the desired partition. In [6], it was shown that the path partition problem is polynomial-time solvable in the class of bipartite distance-hereditary graphs. (The proof uses a similar technique to that of the proof that k-PP is polynomial-time solvable for trees [7].) The k-path partition problem, however remains of unknown complexity on this class. Also, as mentioned in the previous section, the complexity of k-PP is also unknown for the class of biconvex graphs.

References

[1] V. E. Alekseev. On easy and hard hereditary classes of graphs with respect to the independent set problem. *Discrete Applied Mathematics*, 132:17, 2003.

[2] K. Asdre and S.D. Nikopoulos. NP-completeness results for some problems on subclasses of bipartite and chordal graphs). *Theoretical Computer Science*, 381:248, 2007.

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

[4] J. Monnot and S. Toulouse. The path partition problem and related problems in bipartite graphs. *Operations Research Letters*, 35:677, 2007.

[5] G. Steiner. On the k-path partition of graphs. *Theoretical Computer Science*, 290:2147, 2003.

[6] H-G. Yaeh and G.J. Chang. The path-partition problem in bipartite distance-hereditary graphs. *Taiwanese Journal of Mathematics*, 2:353, 1998.

[7] J. Yan, G.J. Chang, S. M. Hedetniemi, and S. T. Hedetniemi. k-path partitions in trees. *Discrete Applied Mathematics*, 78:227, 1997.