k-Product Lists for (Generalized) Feedback Vertex Sets: A Tool for Designing FPT Algorithms*

Rin Saito Yuma Tamura

rin.saito@dc.tohoku.ac.jp tamura@tohoku.ac.jp

Graduate School of Information Sciences, Tohoku University, Sendai, Japan

For combinatorial problems, it is natural to study variants obtained by imposing additional structural constraints on feasible solutions. A standard FPT approach parameterized by the solution size k is to enumerate all inclusion-minimal solutions of size at most k. If this can be done in time $f(k) \cdot n^{O(1)}$, then the number of such solutions is similarly bounded, which often enables efficient algorithms for the corresponding variants. Although such enumeration-based algorithms are sometimes viewed as naive, they frequently provide a useful starting point for more refined techniques.

Given a graph $G = (V, E)$, a *feedback vertex set* is a vertex subset F such that $G[V \setminus F]$ is a forest. The Feedback Vertex Set problem asks whether G admits a feedback vertex set of size at most k. Unlike Vertex Cover, minimal feedback vertex sets cannot be enumerated in FPT time parameterized by k, and hence FPT-time enumeration of all minimal feedback vertex sets is impossible.

To design faster FPT algorithms for Feedback Vertex Set, Guo et al. [1] introduced the notion of a *k-compact list*, an FPT-size data structure that implicitly represents all minimal feedback vertex sets of size at most k. A *compact representation* consists of pairwise disjoint vertex subsets such that choosing one vertex from each subset yields a minimal feedback vertex set. A minimal feedback vertex set is *represented* if it contains exactly one vertex from each subset. A collection of such representation of size at most k forms a *k-compact list* if every minimal feedback vertex set is represented. They showed that a k-compact list can be computed in FPT time, and it has since been used for several variants of Feedback Vertex Set.

Our Contribution. A k-compact list provides an implicit representation of all minimal feedback vertex sets, but it requires that each compact representation generates only minimal solutions. While this restriction is natural from a theoretical perspective, it is often unnecessarily strong for algorithmic applications, where allowing non-minimal solutions can still suffice as long as all minimal ones are covered.

Motivated by this, we introduce a relaxed notion of compact representations. We allow a selection from a representation to yield a non-minimal feedback vertex set, while preserving the key property that every minimal feedback vertex set is represented. We call the resulting structure a *k-product representation*, and a collection of these a *k-product list*.

We show that this relaxed framework remains algorithmically powerful. In particular, it enables FPT algorithms for a wide range of variants of Feedback Vertex Set, including weighted, connected, nonpartisan, hereditary-property, conflict-free, simultaneous, reconfiguration, and diverse variants.

Our contribution is not specific to Feedback Vertex Set, but rather proposes a general data-structural framework for implicitly encoding all minimal solutions. To illustrate its generality, we consider *ℓ-forest deletion sets*, which generalize feedback vertex sets (the case $\ell = 0$). It is known that this problem is FPT parameterized by $k + \ell$ [2, 3]. We show that a k-product list for this generalization can be computed in time $O^*(64^\ell \cdot 17^k)$ for $\ell \geq 0$.[1] As a corollary, we obtain an $O^*(17^k)$-time algorithm for computing a k-product list of feedback vertex sets.

References

[1] Guo, J., Gramm, J., Hüffner, F., Niedermeier, R., Wernicke, S.: Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. Journal of Computer and System Sciences **72**(8), 1386–1396 (2006). https://doi.org/10.1016/J.JCSS.2006.02.001

[2] Lin, M., Feng, Q., Wang, J., Chen, J., Fu, B., Li, W.: An improved FPT algorithm for almost forest deletion problem. Information Processing Letters **136**, 30–36 (2018). https://doi.org/10.1016/J.IPL.2018.03.016

[3] Rai, A., Saurabh, S.: Bivariate complexity analysis of almost forest deletion. Theoretical Computer Science **708**, 18–33 (2018). https://doi.org/10.1016/J.TCS.2017.10.021

*This work was partially supported by JST SPRING Grant Number JPMJSP2114 and JSPS KAKENHI Grant Number JP25K21148.

[1] The O^* notation suppresses polynomial factors.