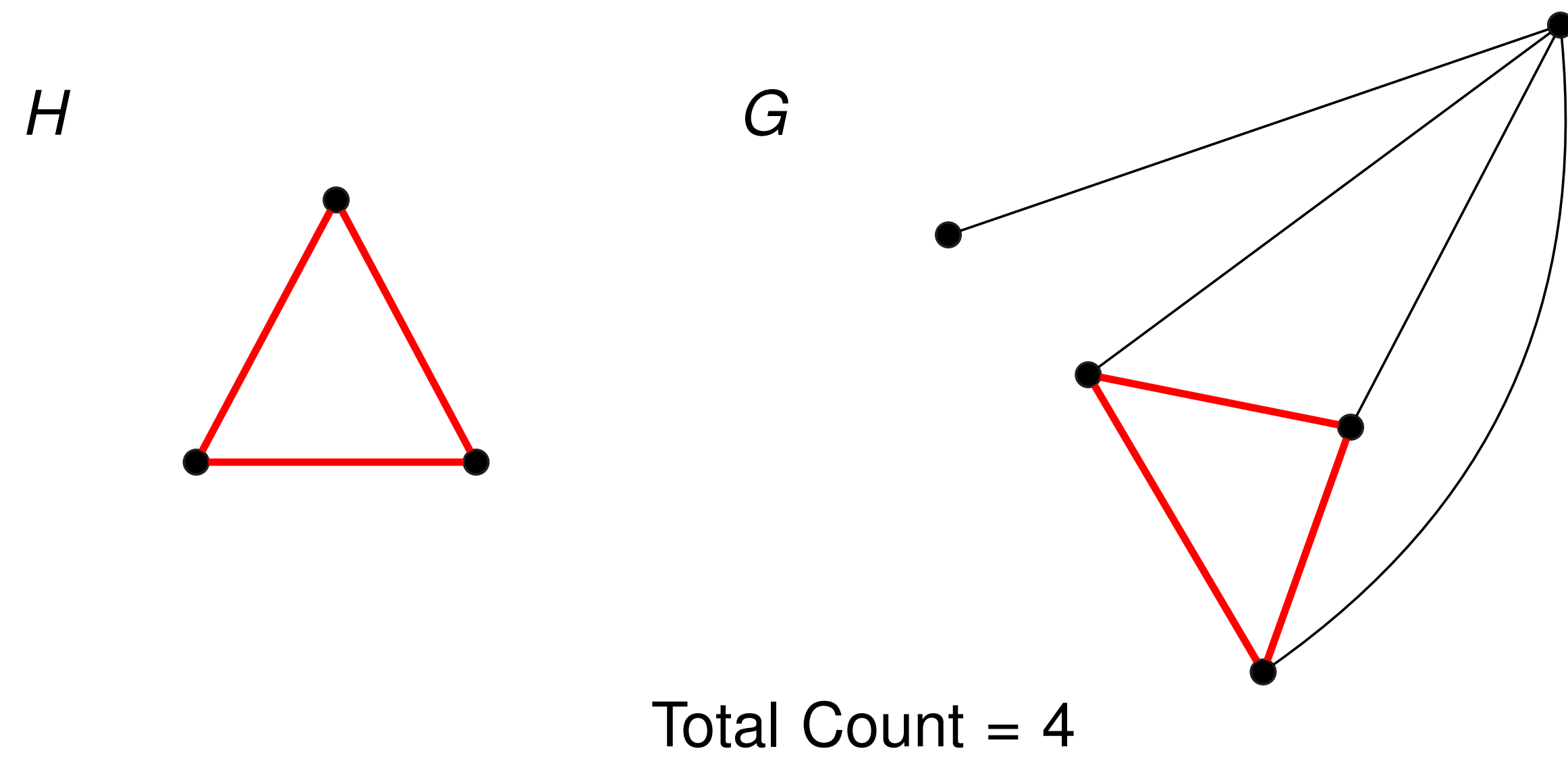




## Key Definitions

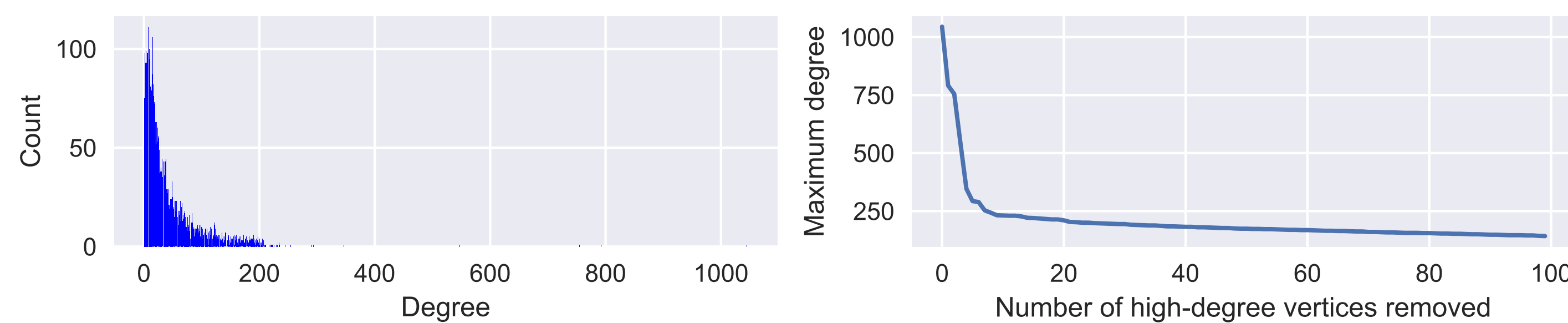
- ▶ The **subgraph counting problem** asks how many unlabelled copies of the graph  $H$  are in the graph  $G$ .



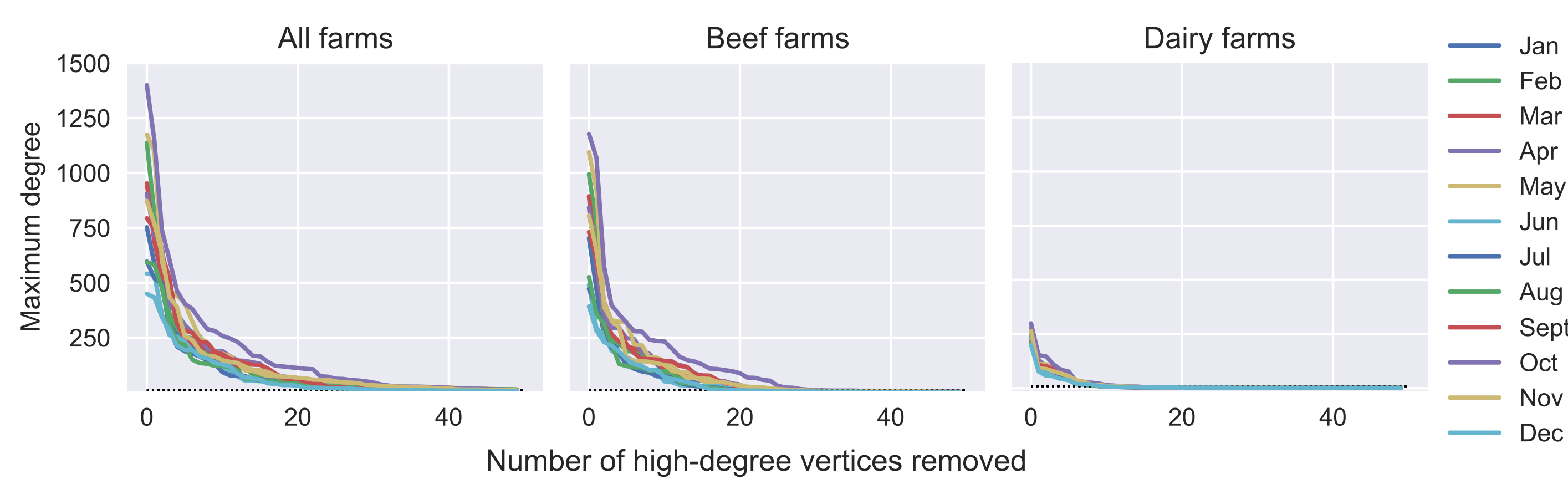
- ▶ A graph has **almost bounded degree** if it has at most  $k$  vertices with degree exceeding  $k$  for some constant  $k$ .
- ▶ A problem is **fixed-parameter tractable (FPT)** if it can be solved by an algorithm in time  $f(k)n^{O(1)}$  where  $n$  is the size of the problem instance,  $k$  is some parameter associated with the problem instance, and  $f$  is a function of  $k$  which is independent of  $n$ .

## Motivation

- ▶ Subgraph counting is a useful way of analysing and comparing real-world networks.
- ▶ Many real-world networks have almost bounded degree:



**Fig. 1:** A plot of the degree distribution of a graph derived from a Facebook data set (left), and the maximum degree of graphs derived from a variety of real-world data sets after greedily removing high-degree vertices (right) [2, 4].



**Fig. 2:** Plots of the maximum degree of graphs derived from the Scottish cattle trading data [2].

## Existing Complexity Results

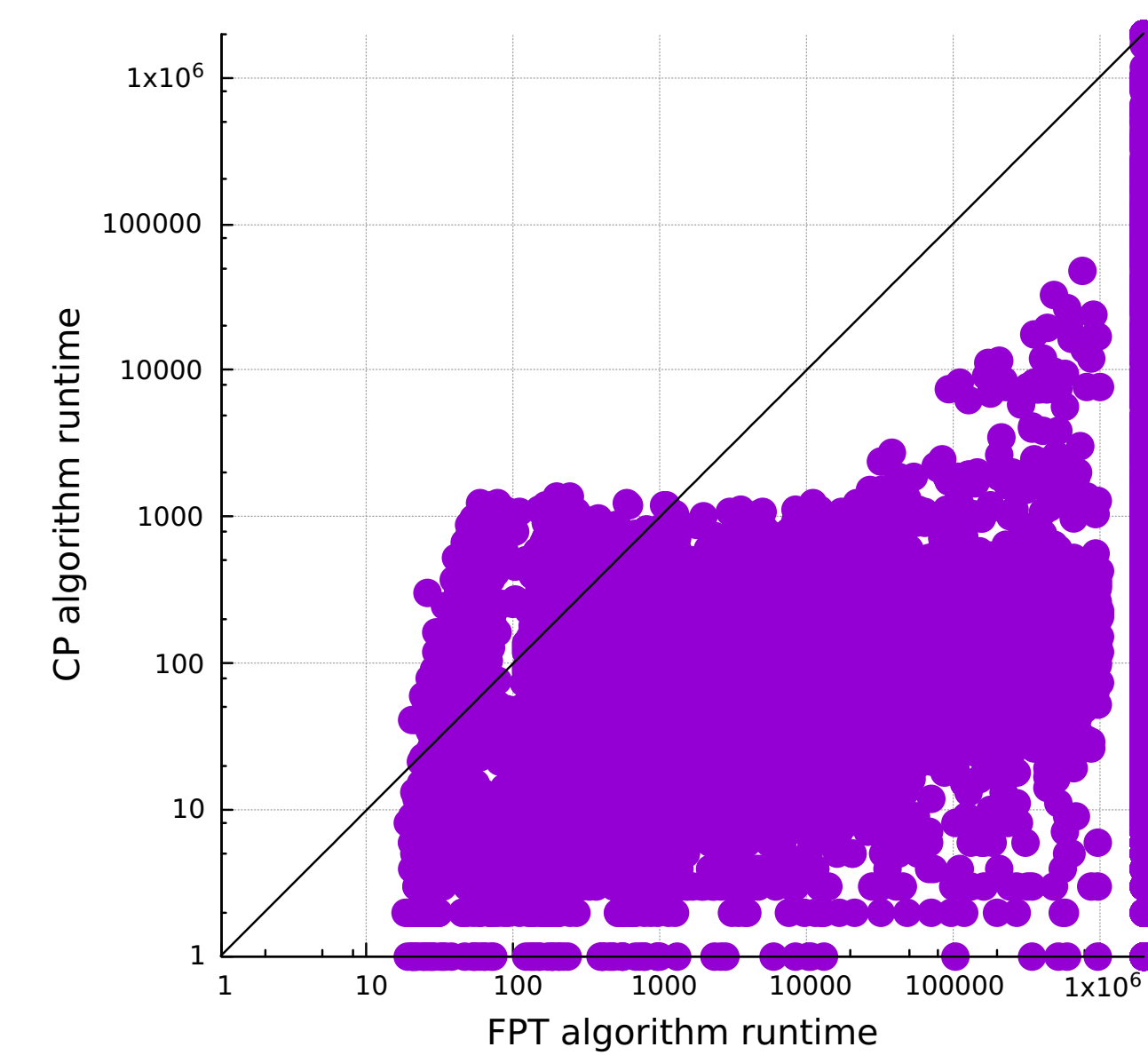
- ▶ Subgraph counting is NP-complete in general.
- ▶ Assuming the Exponential Time Hypothesis, subgraph counting is not in FPT in general [1].
- ▶ Subgraph counting is in FPT for graphs with almost bounded degree [2].

## Project Objectives

- ▶ Write and implement an FPT algorithm for subgraph counting in graphs with almost bounded degree.
- ▶ Test performance of FPT algorithm against more general constraint programming (CP) method [5] on large real-world data sets [3, 6].

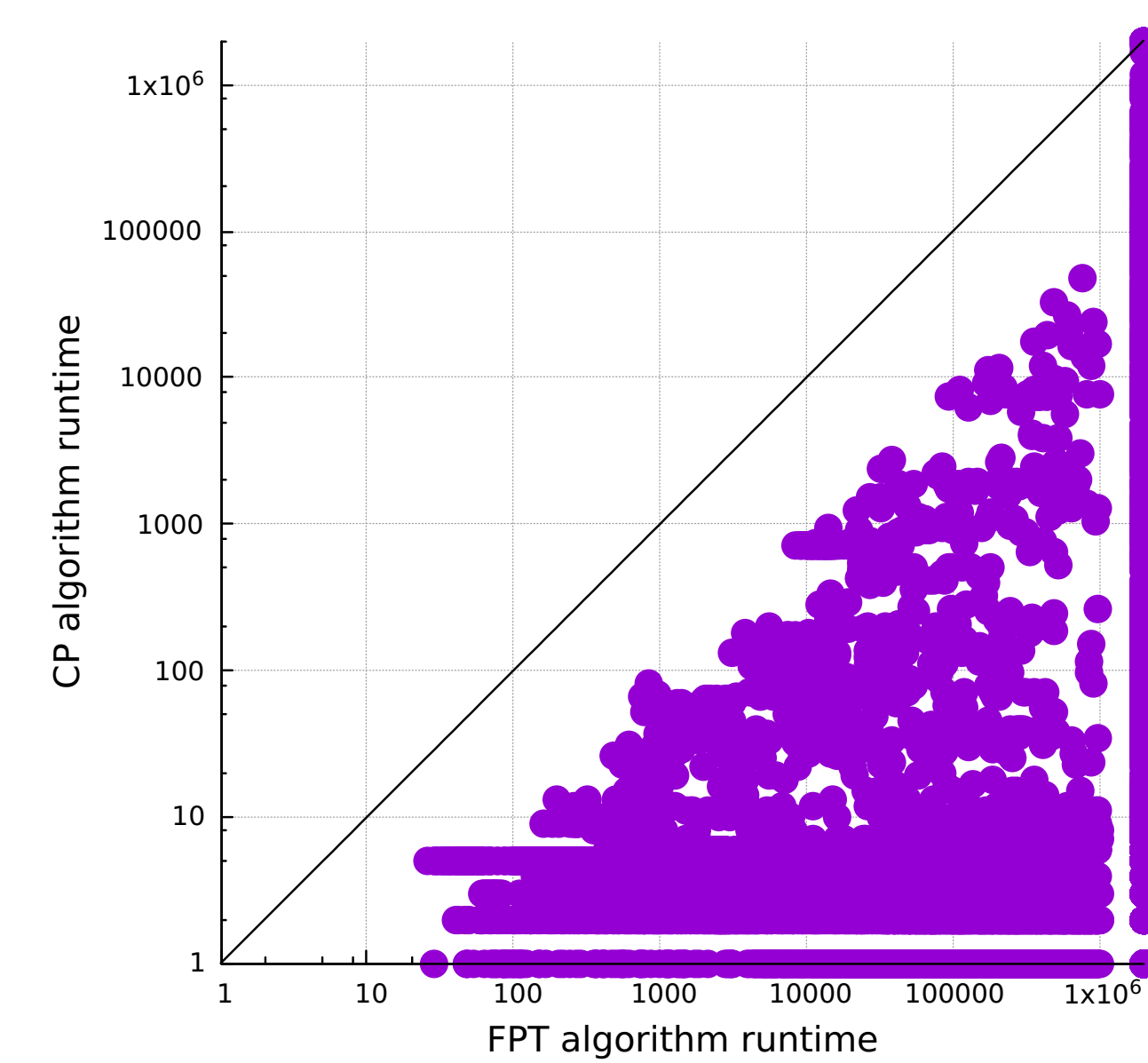
## Conclusions

- ▶ Significant preprocessing carried out by the CP algorithm lead to better performance of the FPT algorithm on “easy” instances (Figure 3).



**Fig. 3:** A plot of the CP algorithm runtime against the FPT algorithm runtime for subgraph counting in real-world networks when the preprocessing time of the CP algorithm is included.

- ▶ When the preprocessing time is excluded, the CP approach outperforms the FPT approach on all instances (Figure 4).



**Fig. 4:** A plot of the CP algorithm runtime against the FPT algorithm runtime for subgraph counting in real-world networks when the preprocessing time of the CP algorithm is excluded.

## Future Work

- ▶ Combine FPT approach with improved search methods.
- ▶ Run experiments on further data sets known to have almost bounded degree.
- ▶ If the combined approach is successful, develop a per-instance algorithm selection model for subgraph counting.

## References

- [1] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. System Sci.*, 72(8):1346–1367, 2006.
- [2] Jessica Enright and Kitty Meeks. Counting small subgraphs in multi-layer networks. *CoRR*, abs/1710.08758, 2017.
- [3] Lars Kotthoff, Ciaran McCreesh, and Christine Solnon. Portfolios of subgraph isomorphism algorithms. In *International Conference on Learning and Intelligent Optimization*, pages 107–122. Springer, 2016.
- [4] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014.
- [5] Ciaran McCreesh and Patrick Prosser. A parallel, backjumping subgraph isomorphism algorithm using supplemental graphs. *Lecture Notes in Computer Science Principles and Practice of Constraint Programming*, page 295–312, 2015.
- [6] Christine Solnon. Benchmarks for the subgraph isomorphism problem. <https://perso.liris.cnrs.fr/christine.solnon/SIP.html>.