# Between Subgraph Isomorphism and Maximum Common Subgraph

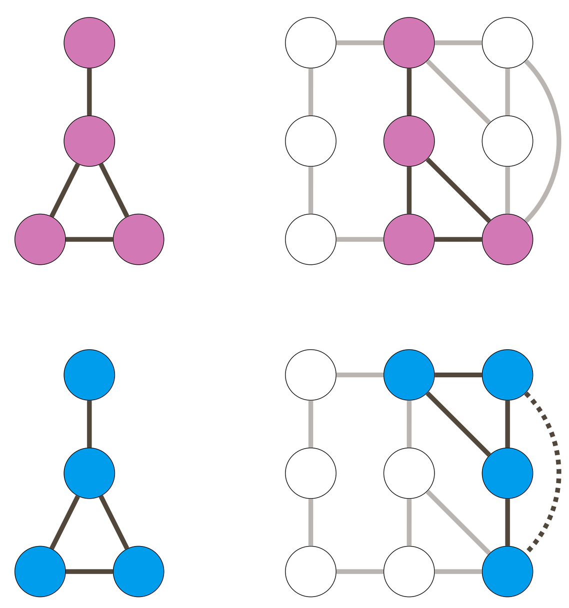
**Ruth Hoffmann, Ciaran McCreesh and Craig Reilly**, University of Glasgow, Glasgow, Scotland

When a small pattern graph does not occur inside a larger target graph, we can ask how to find **"as much of the pattern as possible"** inside the target graph. In general, this is known as the **maximum common subgraph problem**, which is much more computationally challenging in practice than **subgraph isomorphism**. We introduce a restricted alternative, where we ask if all but $k$ vertices from the pattern can be found in the target graph. This allows for the development of slightly weakened forms of certain invariants from subgraph isomorphism which are based upon degree and number of paths. We show that when $k$ is small, weakening the invariants still retains much of their effectiveness. We are then able to solve this problem on the standard problem instances used to benchmark subgraph isomorphism algorithms, despite these instances being too large for current maximum common subgraph algorithms to handle. Finally, by iteratively increasing $k$, we obtain an algorithm which is also competitive for the maximum common subgraph problem.

## Subgraph Isomorphism (Finding Patterns in Graphs)

The **non-induced subgraph isomorphism problem** is to find an injective mapping from a given pattern graph to a given target graph which preserves adjacency.

The **induced** variant of the problem additionally requires that the mapping preserve non-adjacency, so there are no "extra edges" in the copy of the pattern that we find. The top example is induced, whereas the bottom example is not, due to the dashed edge.

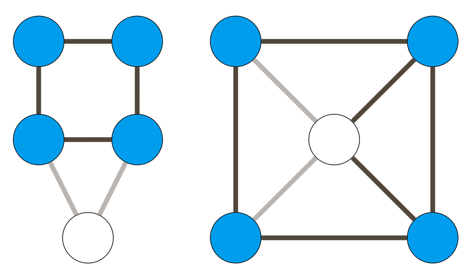


## Applications

Despite these problems being NP-complete, modern practical subgraph isomorphism algorithms can handle problem instances with many hundreds of vertices in the pattern graph, and up to **ten thousand vertices** in the target graph, leading to successful application in areas such as **computer vision**, **malware detection**, **compilers**, **model checking**, **biochemistry**, and **pattern recognition**.

## Maximum Common Subgraph (Comparing Graphs)

The **maximum common subgraph problem** is to find the largest graph which is isomorphic to a subgraph of two graphs simultaneously—the size of the maximum common subgraph gives us a measure of how similar two graphs are.



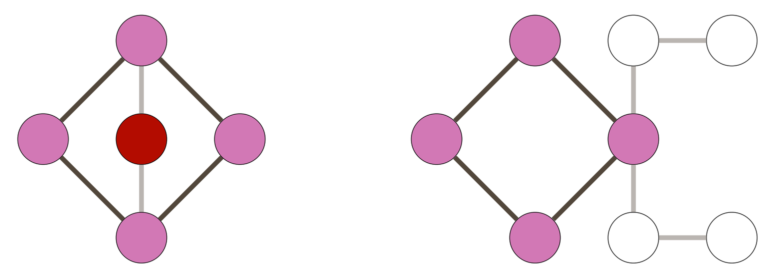
In the induced variant of subgraph isormorphism when a pattern cannot be found in a target graph, we may seek a result which maps as many vertices of the pattern into the target as possible. This is precisely the maximum common subgraph problem.

The maximum common subgraph problem is much more difficult in practice than subgraph isomorphism. The state of the art for the maximum common subgraph problem becomes **computationally infeasible at only 35 vertices** when working with unlabelled graphs. This is largely because strong inference, based upon the degrees of vertices and the distances or paths between them, is possible with subgraph isomorphism, but not maximum common subgraph.

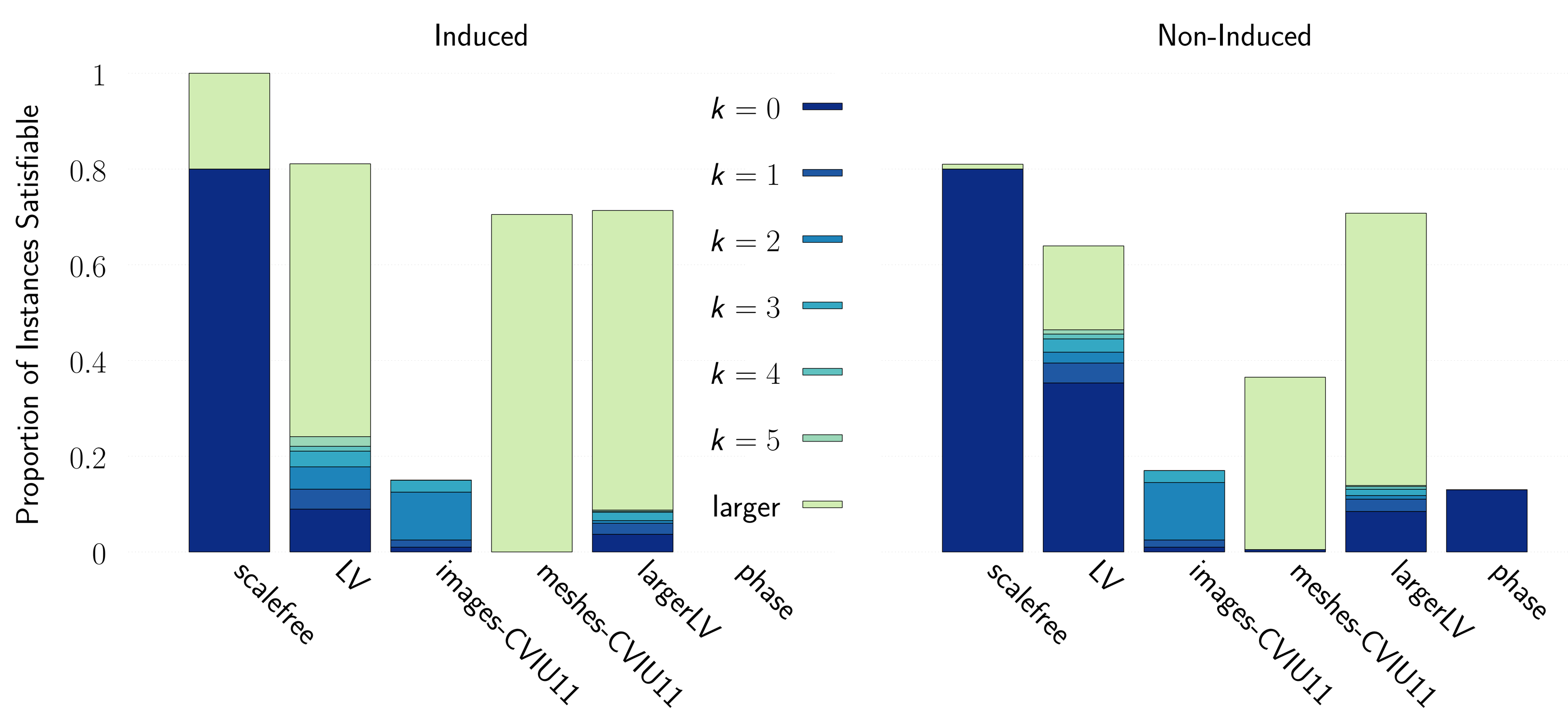
## $k$-less Subgraph Isomorphism (Most of a Pattern)

A **$k$-less subgraph isomorphism** between a pattern and target graph, is a subgraph isomorphism where we seek a mapping from all but $k$ verticies of the pattern to the target.

If $k$ is reasonably small, weakened forms of invariants from subgraph isomorphism (based on vertex degrees and paths) are still effective at reducing the search space. In the example, upon removal of the central vertex from the pattern, we have a $1$-less isomorphism from the pattern to the target.



Are there instances for which $k = 0$ is unsatisfiable, but that are satisfiable for small $k$? For the problem families which do not consist entirely of satisfiable instances, we plot this for both the induced and non-induced variants.



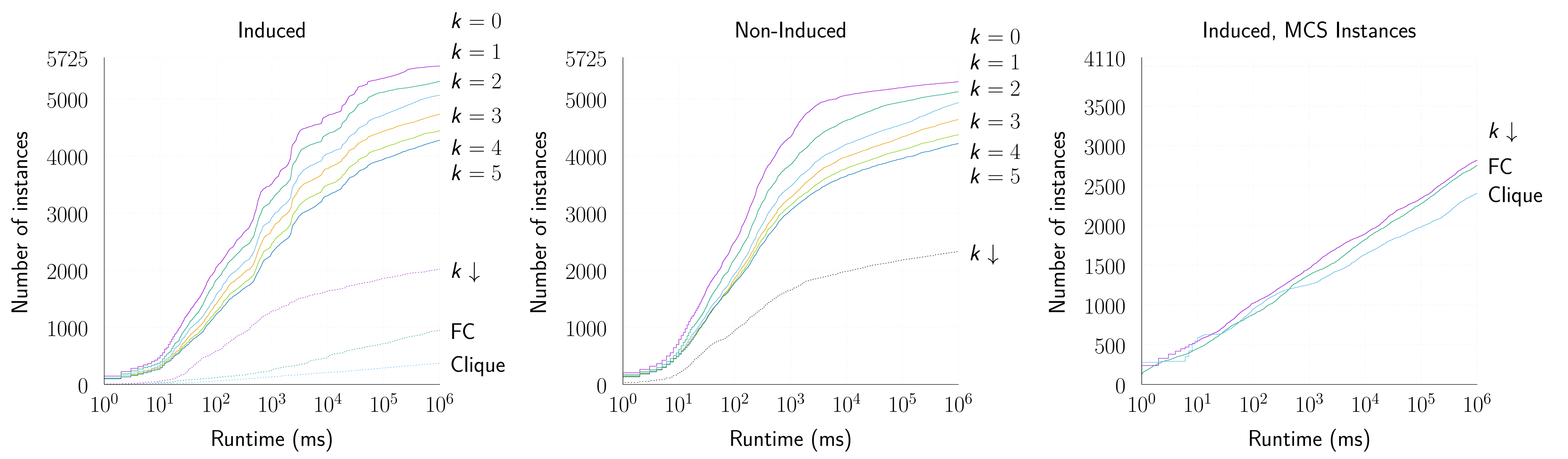

In the "phase" family, which consists of instances crafted to be extremely difficult to solve, we are not able to answer this question, and in the "scalefree" family we see no satisfiable instances with low but not zero $k$.

However, in several of the remaining families our algorithm manages to provide exact solutions for many instances. This is particularly interesting for the "images-CVIU11", where the size of the solution has a direct **real-world interpretation** in terms of **closeness of image matching**.

## Maximum Common Subgraph: Solving from the Top Down

What would happen if we used the $k$-less approach to solve the maximum common induced problem? We could simply start at $k = 0$, and increase $k$ until a solution is found. This **tackles the problem in the opposite direction to existing approaches**, which work by attempting to construct larger and larger solutions. In the figure, "$k\downarrow$" is our algorithm, whilst "FC" and "clique" are state of the art maximum common subgraph algorithms.

We see that **our approach is able to close over twice as many of these instances as the previous state of the art**—which struggles due to its RAM requirements. On instances designed for the maximum subgraph problem we are still the single strongest solvers, and our performance tends to be complementary to that of the clique encoding.




## See the Paper for...

- Description of our new invariants, and proofs that they hold.
- Description of the algorithm.

## Future Work

Develop an algorithm for maximum common subgraph which makes use of an upper bound from our approach and a lower bound from the clique approach, stopping when the bounds converge.

This work was supported by the Engineering and Physical Sciences Research Council [grant numbers EP/K503058/1 and EP/M506539/1].

c.reilly.2@research.gla.ac.uk