Twin-Width Fuzzification

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Marek Effenberger [1] Supervisor: doc. RNDr. Dana Hliněná, Ph.D. [1]

[1]Brno University of Technology

Excel@FIT 2025

Abstract

Twin-width is a graph complexity measure introduced by Bonnet et al. in 2020 [1]. Essentially, it is a single integer that describes the structural complexity of any given graph using sequences of vertex contractions. The use of this parameter proved beneficial, mainly in the field of parametrized complexity theory. As of yet, the measure has not been described on fuzzy graphs. These graphs possess a better descriptive ability, as we can model uncertain relations among the vertices, mapping both the set of vertices and the set of edges on a real interval. This work shows an approach to fuzzify the twin-width to this graph type, presents its multiple properties and discusses the boundedness of this parameter over respective fuzzy graph families.

Twin-Width

If we consider a graph to have its edges composed of two disjoint sets, red and black, we speak of a trigraph. On the initial graph, we perform a contraction of any two vertices u, v. This contraction constructs a new trigraph, which does not contain u, v vertices but a new vertex w. The vertex w has to be connected to the union of neighbours u and v. For the neighbours x we color these edges according to these rules [1]:

- If u and v share a black edge with x in the original graph, the new edge connecting w and x is black,
- In any other case, for example, if only one vertex u is connected to x, the edge becomes red.

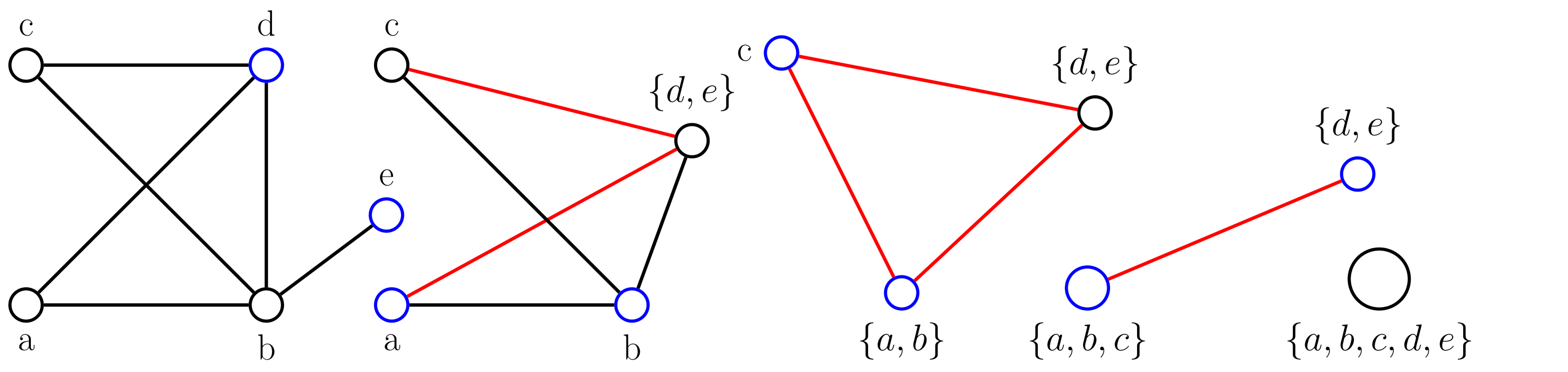

Figure 1. 2-sequence of a Graph G

In Figure 1 we contract the first two vertices d and e in the input leftmost graph. As both were connected to b, the new vertex $\{d, e\}$ in the next graph shares a black edge with b. With a and c, only d shared an edge. Therefore, the new edges become red.

The input graph is contracted until trigraph of one vertex remains. The width of a sequence is the maximum number of red neighbours among all vertices of every graph within the sequence. In Figure 1, the width is $w = 2$ as $\{d, e\}$ has two red neighbours and there is no vertex in any graph of the sequence having more [1].

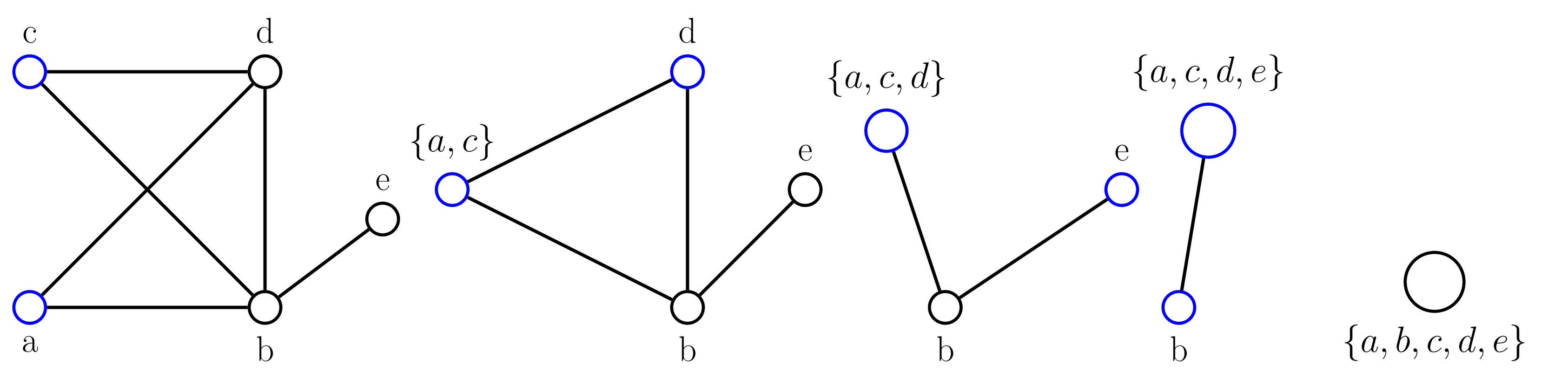

Figure 2. Optimal Contraction Sequence of G

If we construct all the possible contraction sequences of the input graph and select the one with the minimum width, then this sequence is considered optimal. This width is considered the twin-width of the input graph. In Figure 2 we see an optimal sequence for the graph in Figure 1, its twin-width is equal to $tw = 0$.

Computation Algorithm

The algorithm utilised for computing twin-width is outlined as follows:

```
Algorithm 1: Brute-Force Algorithm
  Input: Graph Graph, Max-Degree maxDeg, Merge Sequence Seq
  Output: Fuzzy Twin-Width ftw, Optimal Sequences OptSeq
  Function FuzzyTW(Graph, maxDeg, Seq):
    if len(Graph.vertices) = 1 then
      if maxDeg < ftw then
        ftw ← maxDeg;
        OptSeq ← [(Seq)];
      if maxDeg = ftw then
        OptSeq.append(Seq);
      return;
    else
      foreach (u, v) ∈ Graph.vertices do
        newGraph ← Graph.merge(u, v);
        maxError ← newGraph.findMaxErr();
        newSeq ← Seq + [(u, v)];
        FuzzyTW(newGraph, max(maxDeg, maxError), newSeq);
```

Figure 3. Brute-force Algorithm

The complexity of this algorithm is $2^{\mathcal{O}(n \cdot log(n))}$, which is hard. However, the twin-width can be analytically bounded for multiple graph families, such as paths or cycles.

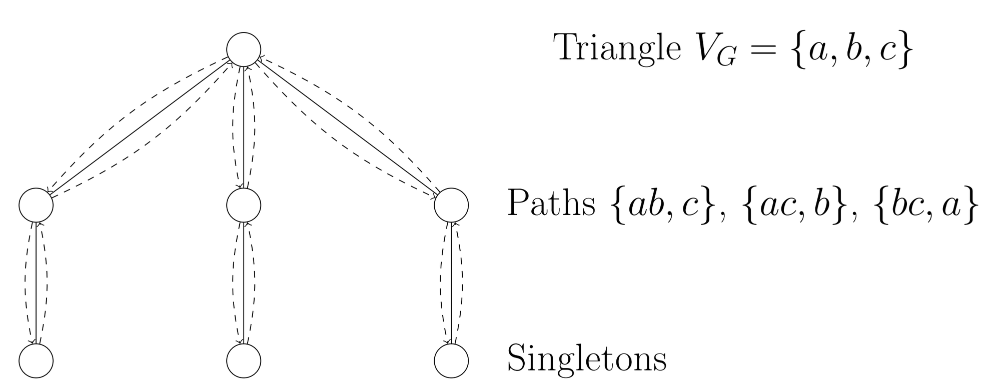

Figure 4. Depth-First Search on Input Triangle

The input graph is effectively processed in a DFS manner as seen in Figure 4. For each contraction, the algorithm searches for the maximal error degree and passes this value further. After the graph is contracted to a singleton, the computed width is compared with the minimum observed among the sequences.

Fuzzy Graphs

Commonly denoted as $G = (V, \sigma, \mu)$, where [2]:

- V is a set of vertices,
- σ is a vertex membership function – $\sigma \mid V \to [0, 1]$,
- μ is an edge membership function – $\mu \mid \mathcal{E} \to [0, 1]$.

The $[0, 1]$ notation means that the element is assigned a value on the real interval from zero to one. The edge membership function is constrained by $\mu(x, y) = \otimes(\sigma(x), \sigma(y))$, where $\otimes$ denotes a binary function called t-norm. Most research concerns the fuzzy graphs constrained by the minimum t-norm.

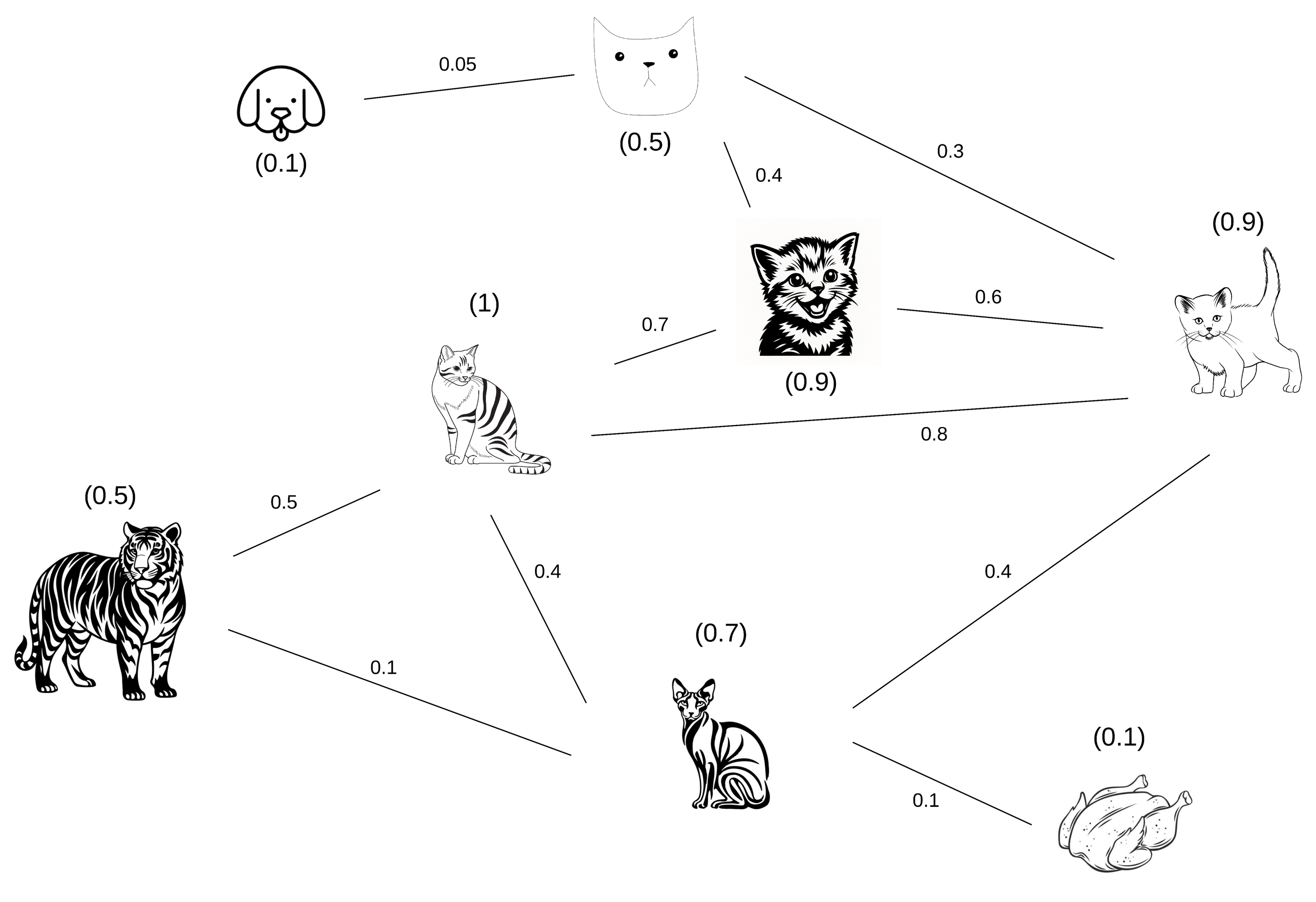

Figure 5. Fuzzy Graph

One example can be seen in Figure 5. Imagine a situation where you searched for cats. The search engine retrieved some images, and each image can be evaluated using an assumption "How much is this cat really a cat?". The retrieved images share some similarities, but each pair of cats shown can share a relation maximally valued at the minimum of the catness of both cats.

Fuzzy Twin-Width

The fuzzification process has to account for the partial relations. The approach concerns partial redness and blackness of a degree, computed using a combination of triangular norms and triangular conorms.

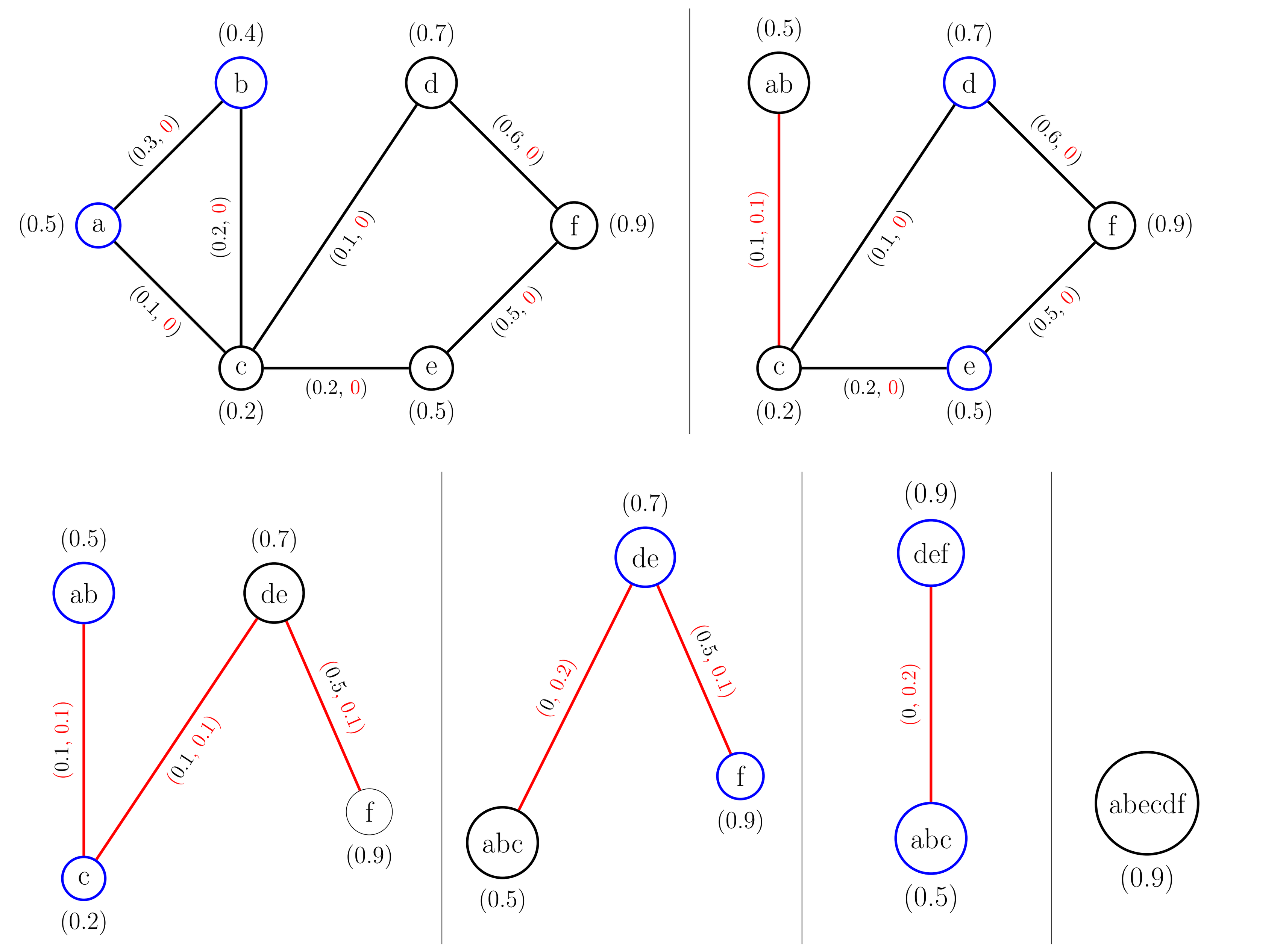

Figure 6. Contraction Sequence For G_6

These functions model fuzzy intersection (t-norms) and fuzzy union (t-conorms). The total edge is computed using the fuzzy union, while its blackness amounts to the value of the fuzzy intersection. The redness is computed as the subtraction of the total edge value from its blackness.

As seen in Figure 6, the redness is on the real domain. Therefore, the minimum of the maximal summations of red neighbours gives us a real value of fuzzy twin-width as well. The work facilitates the fuzzy twin-width properties and discusses its boundedness on certain fuzzy graph families.

Achievement Summarization

The expansion of the twin-width to fuzzy graphs proves useful, as effectively, any data in the form of a fuzzy graph can have its fuzzy twin-width measure computed, gaining a new insight into its structural complexity in terms of its lossy compression. As the computation remains hard, the work also discusses bounding of several fuzzy graph families, which shall speed up this computation by several orders. A part of this thesis is a website-based application used for the computation simulation and vertex merging visualisation hosted on https://fuzzy-twin-width.onrender.com/. The website allows users to construct and adjust any fuzzy graph constrained by four basic triangular norms and run the fuzzy twin-width computation.

References:

[1] Bonnet, É. Twin-Width and Contraction Sequences. 2024. Habilitation à diriger des recherches. École Normale Supérieure de Lyon. Available at: https://perso.ens- lyon.fr/edouard.bonnet/text/hdr.pdf

[2] Mathew, S.; Mordeson, J. N. and Malik, D. S. Fuzzy Graphs. In: Fuzzy Graph Theory. Cham: Springer International Publishing, 2018, p. 1383. ISBN 978-3-319- 71407-3. Available at: https://doi.org/10.1007/978-3-319-71407-3_2

xeffen00@stud.fit.vutbr.cz