BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2025
http://excel.fit.vutbr.cz

Twin-Width Fuzzification

Marek Effenberger*

Abstract
Twin-width is a graph complexity measure which utilises sequences of vertex contractions to describe the structural complexity of any given graph using a single integer. Fuzzy graphs extend the notion of crisp graphs by assigning each vertex and edge a membership value on a real interval representing various degrees of uncertainty. As of the writing of this thesis, no attempts have been made to expand the twin-width notion to fuzzy graphs. The primary result of this work is the fuzzification of twin-width. This work facilitates the fuzzification process, emphasises its properties and discusses the boundedness of the newly presented measure, fuzzy twin-width, within certain fuzzy graph families.

*xeffen00@stud.fit.vut.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Many computationally difficult problems are defined on graphs. One of the approaches to reduce this complexity is to look at the structure of the graph on which the problem is defined. There are multiple graph complexity measures that capture this complexity using a single integer. For certain problems, this number can be treated as a special input (a parameter) when we talk about how long the computation takes (using big-$\mathcal{O}$ notation). The part of the computation that becomes very large very quickly (the 'combinatorial blow-up') is limited by this parameter, while the graph size only increases the computation time at a more manageable rate (polynomially, linearly,...).

2. Computation of Twin-Width

One of these measures is called twin-width. Twin-width is a value computed by algorithmically merging the input graph. The algorithm works by choosing any two vertices u, v, merging them and constructing a new graph whose set of vertices lacks both u and v but includes a new vertex w. In this graph, the neighbouring vertices connected to w are those that neighbored with u and v in the previous graph. The edges from w to the neighbour x are then described according to these rules [1]:

- The edge becomes black if both u and v shared an edge with x,
- The edge becomes red in any other situation.

The new graph with red edges (also known as trigraph) is then merged in the same manner. This process is applied recursively until a graph composed of only one vertex is left [1].

In the resulting sequence of graphs, we look at all the vertices and find the one with the most neighbours connected via a red edge. The number of these red neighbours is known as the width of the sequence. If we construct all the possible contraction sequences, compute their width and select the minimum of these values, we get the twin-width of the input graph [1].

In the first Figure, we can observe a contraction sequence of a graph of five vertices, which yields a width of two. However, in Figure 2, we see another way of merging the vertices, which results in a width of zero, which is the twin-width of the graph.

Effectively, the algorithm for brute-force computation processes the input graph in a DFS manner, merges the vertices, and computes the width of the sequences as seen in Figures 3 and 4. The complexity of this algorithm is high. If we want to find the measure of any graph, the brute-force method runs in $2^{\mathcal{O}(n \cdot log \cdot n)}$. However, in many families, the measure is analytically bound. For example, a path of four or more vertices possesses a twin-width of one, and any tree can have its twin-width valued maximally at two. To compute whether the input graph belongs to some family in which the twin-width value is bounded is often several orders less complex than the brute-force algorithm.

3. Fuzzy Graphs

To fuzzify this measure, we first need to describe fuzzy graphs. These graphs are able to express uncertain relations between their elements. Each vertex composing the fuzzy graph possesses a membership function σ. This function maps the vertex to the real interval [0,1]. The edges of these graphs possess an edge mapping membership function μ that maps an edge to the same interval of $[0,1]$ while being upper bounded by $\mu(x,y) \leq \otimes(\sigma(x),\sigma(y))$ [2]. The symbol $\otimes$ denotes a triangular norm that is a binary function $[0,1]\times[0,1]\to[0,1]$. The most common t-norm used in fuzzy graphs for the μ constraint is the minimum t-norm.

If we look at the Figure 5 depicting a fuzzy graph of cats, each image of a cat (a vertex) is mapped to the real interval $[0,1]$. Each pair of cats is connected by an edge that is maximally valued at the minimal value of σ of the two images it connects. For instance, if we examine the sphynx cat and the image of a frozen chicken, we have $\sigma(sphynx) = 0.7$ and $\sigma(chick) = 0.1$. The edge that connects them is valued at the minimum of these two, which gives $\mu(sphynx, chick) = \min(0.7, 0.1) = 0.1$.

4. Fuzzification Process

To fuzzify the twin-width algorithm it is important to account for the partial belonging to a set. Therefore, there is an importance not to simply mark some edges as red, but to allow the fuzzy (tri)graphs to possess a partial level of redness.

The designed process utilises triangular norms and triangular conorms. These functions are commonly used to model fuzzy intersection and fuzzy union, respectively. For example, if we merge two vertices in a fuzzy graph, the total value of μ, denoted μ_T, of the new vertex and some neighbour, is calculated using the t-conorm of the μ_T of the two edges which connect the merged vertices to this neighbour. From the value of μ_T we derive two components – μ_R, which is the redness, and μ_B, which is the blackness of an edge. The blackness is computed using the t-norm of blackness of the two edges while the redness is computed using the subtratction $\mu_T - \mu_B$. The process is depicted in Figure 6.

In this manner, we construct all the possible contraction sequences, select the vertex of each sequence with the most redness connected to it, and finally, select the sequence of the lowest width, marking it optimal, making the width value the final value of the fuzzy twin-width of the input fuzzy graph.

The computation of fuzzy twin-width does not differ much from the brute-force algorithm, as the calculation of the binary functions adds constant computational complexity. In every other way, the algorithm remains the same. But, just as for the crisp graphs, analytically bounding the value of fuzzy twin-width is possible in certain fuzzy graph families. However, this bounding is more complex as the value of twin-width depends on each edge's value of μ, and differs by using different t-norm calculations. This work discusses the bounding of fuzzy paths, fuzzy cycles, fuzzy trees and complete fuzzy graphs.

If we consider data which can be modelled using a fuzzy graph, the fuzzy twin-width might be considered a measure of its complexity in regard to lossy compression.

The observations of the optimal sequences and the contraction process were made using a website-based application. This application allows users to create and modify a fuzzy graph while employing the brute-force method for fuzzy twin-width calculation.

5. Conclusions

The work presents a mathematically sound way to describe fuzzy graphs using a single integer by running the fuzzy twin-width computation. Furthermore, the work discusses the parameter's boundedness over certain fuzzy graph families and presents a website application, which can run the brute-force algorithm on any input fuzzy graph. The apparatus presented might be used to describe vague systems in terms of lossy compression.

Acknowledgements

I want to thank my supervisor, doc. RNDr. Dana Hliněná, Ph.D., for her help with the formulation of the theory, guidance and kindness. I would also like to express my gratitude to prof. RNDr. Petr Hliněný, Ph.D. and Ing. Petr Veigend, Ph.D, for their help with the thesis refinement.

References

[1] Édouard Bonnet. *Twin-Width and Contraction Sequences*. Habilitation à diriger des recherches, École Normale Supérieure de Lyon, 2024.

[2] Sunil Mathew, John N. Mordeson, and Davender S. Malik. *Fuzzy Graphs*, pages 13–83. Springer International Publishing, Cham, 2018.