Measuring the extent of interdisciplinary research and creating a collaboration group structure at KTH

A study using a flow-based clustering algorithm developed by Flake et al.

AGNES ÅMAN

HANNA NYBLOM
Measuring the extent of interdisciplinary research and creating a collaboration group structure at KTH

A study using a flow-based clustering algorithm developed by Flake et al.

Att undersöka utbredningen av den interdiciplinära forskningen och ta fram samarbetsgrupper på KTH

En studie genomförd med användning av den flödesbaserade klustringsalgoritmen utvecklad av Flake et al.

AGNES ÅMAN
HANNA NYBLOM

Degree Project in Computer Science, DD143X
Supervisor: Arvind Kumar
Examiner: Örjan Ekeberg

CSC, KTH, May 11, 2016
Abstract

With interdisciplinary research being a possibility in modern research environments, it is interesting to optimise collaborations between researchers in order to further develop the research environment. The scope of this thesis was therefore to develop a method to measure how widespread the interdisciplinary research is and to propose collaboration groups of researchers created by the use of graph theory.

This problem was approached by studying the research at KTH by collecting research publications from the publication database DiVA using a web crawler. Then representing the authors of the publications as nodes and the collaborations between two authors as edges in a graph. A graph partitioning algorithm developed by Flake et al. was chosen after a literature study, then applied to the graph to produce the requested collaboration groups.

The results showed that while interdisciplinary research is not the norm at KTH, 23% of the proposed collaboration groups consisted of two or more researchers from different schools at KTH. The original ratio of school association was retained through the partitioning of the graph. A measurement of collaboration per researcher in each collaboration group was suggested and the calculated values of these measurements was found to be largely in the same range, with the exception of one collaboration group. The results also highlighted some inconsistencies in DiVA.

The conclusions were that interdisciplinary research was not very widespread at KTH, however 77 groups were suggested which could be of use for researchers at KTH from now on and in the future. A conclusion was also that this method for finding suitable collaboration groups could be applied at other universities where perhaps interdisciplinary research is more frequent.
Sammanfattning

Eftersom interdisciplinär forskning är förekommande bland dagens forskning är det intressant att optimera och förenkla samarbete mellan forskare för att främja utvecklingen av forskningsmiljön. Målet med denna uppsats var därför att undersöka hur utbredde den interdisciplinära forskningen är och föreslå passande samarbetsgrupper för forskare framtagna genom att använda grafteori.

Avhandlingar skrivna av forskare på KTH samlades ihop från publikationsdatabasen DiVA med hjälp av en så kallad web crawler. Dessa avhandlingar sammanställdes sedan i en graf genom att representera författare som noder och samarbeten som kanter. En grafpartitioneringsalgoritm valdes efter en genomförd litteraturstudie. Denna grafpartitioneringsalgoritm applicerades sedan på grafen och producerade de sökta samarbetsgrupperna.

Resultaten visade att även om interdisciplinär forskning inte är norm på KTH så innehöll 23% av de föreslagna samarbetsgrupperna två eller fler forskare från olika skolor på KTH. Den ursprungliga fördelningen av skolanknytning var bibehållen även efter partitioneringen av grafen. Ett mätt för samarbeten per forskare i varje samarbetsgrupp föreslogs och de flesta av dessa värden visade sig ligga i samma spann, med undantaget av en samarbetsgrupp. Dessa resultat visade på inkonsekvent inmatning av data i DiVA.

Sammanfattningsvis fastställdes det att interdisciplinär forskning på KTH ännu inte är så utbrett. 77 samarbetsgrupper som i framtiden skulle kunna användas av forskare på KTH föreslogs. En slutsats var även att denna metod för att hitta passande samarbetsgrupper skulle kunna användas hos andra universitet där interdisciplinär forskning är vanligare.
## Contents

1 **Introduction**  
   1.1 Problem statement ........................................... 1

2 **Background**  
   2.1 Clustering algorithms ......................................... 2  
   2.2 Earlier work .................................................... 3  
      2.2.1 Clustering based on citations ............................. 3  
      2.2.2 Studies by Flake et al. ................................. 3  
   2.3 Research at KTH .............................................. 4  
   2.4 Graph Theory .................................................. 4  
      2.4.1 Flow Network ............................................ 4  
      2.4.2 Karzanov’s Preflow ...................................... 5  
      2.4.3 Minimum cut, maximum flow ........................... 5  

3 **Methods**  
   3.1 DiVA .......................................................... 6  
   3.2 Web crawler .................................................... 6  
   3.3 Database ....................................................... 7  
   3.4 Graph clustering algorithm ................................. 7  
      3.4.1 Gomory-Hu ............................................... 8  
      3.4.2 Gusfield ............................................... 8  
      3.4.3 Push-relabel ........................................... 8  
   3.5 Implementation of algorithm ............................... 11  
      3.5.1 Graph data structure .................................. 11  
      3.5.2 Extracting components ................................. 11  
      3.5.3 Choosing $\alpha$ ...................................... 11  
      3.5.4 Choosing group sizes ................................. 12  
      3.5.5 Returning clusters .................................... 12  

4 **Result**  
   4.1 Collected data from DiVA ................................. 12  
   4.2 Collaboration groups ........................................ 13  
      4.2.1 School representation in collaboration groups .... 16  
      4.2.2 Collaborations per researcher ........................ 17  

5 **Discussion**  
   5.1 Interdisciplinarity in collected data ..................... 19  
   5.2 Evaluation of collaboration groups ....................... 19  
   5.3 Limitations of DiVA ......................................... 20  
   5.4 Further studies and possible improvements .............. 21  

6 **Conclusion** ................................................... 22
1 Introduction

Universities today are traditionally divided into schools and departments according to different research fields. This structure does not take into account the interdisciplinary research between different departments or different schools.

Oxford dictionaries define interdisciplinary as “Relating to more than one branch of knowledge”[4]. By defining interdisciplinary research as research between researchers from different departments and schools the degree of interdisciplinarity can be measured by looking at actual research collaborations. By choosing a specific university it would then be possible to evaluate how widespread the interdisciplinary work currently is at that university.

The idea for this thesis is to define a collaboration as two researchers being co-authors on a research publication. The aim is then to measure the proportion of the collaborations that are between different departments or between different schools to be able to investigate the interdisciplinary at the chosen university.

The goal is thereafter to find clusters of researchers collaborating with each other to be able to suggest a structure of collaboration groups that does not depend on the current department and school structure. This structure could hopefully be added to the current structure to benefit the researchers collaborating with each other and enhance the research environment at the chosen university.

The idea is to display the collaborations between researchers at a specific university as a graph, with the researchers as nodes and the edges as the collaborations between them with the weights being the number of times they have collaborated. A clustering algorithm suitable for this graph can then be implemented to find collaboration groups reflecting the current collaboration landscape. The algorithm chosen for this thesis is a flow based graph-partitioning algorithm developed by Flake et al.[12].

The university chosen for this project is KTH Royal Institute of Technology, a university organised with 10 schools that is each divided into different departments [26]. By performing this study with publication data from KTH the aim is to evaluate the chosen method so that it can be further developed in future studies and possibly used with data from other universities with similar structure.

1.1 Problem statement

- How widespread is the interdisciplinary research currently at KTH?
- Is it possible to find collaboration groups that reflects the current research landscape by implementing the algorithm developed by Flake et al.?
- How interdisciplinary are these collaboration groups?
2 Background

Since the goal is to represent the researchers and their collaborations as a graph and make use of a clustering algorithm in order to partition these researchers into groups, a literature study was carried out, exploring the available variants of clustering algorithms. A literature study was also conducted to identify earlier work similar to the work presented in this thesis. Finally the theory of flow networks was studied.

2.1 Clustering algorithms

There exists a large number of different clustering algorithms, all with their own strengths, weaknesses and purposes. Based on the natures of generated clusters, techniques and theories behind them, a division into the following four categories of clustering algorithms can be made:

- Hierarchical
  Connectivity clustering or hierarchical clustering connects objects to form clusters based on the objects distance. Objects which are closer together are considered to be more related and vice versa. The distances between objects can be computed in a number of different ways and that is the reason for the wide variety of connectivity-based clustering algorithms.
    - Agglomerative
      Single linkage, complete linkage, group average linkage, median linkage, centroid linkage, Ward’s method, balanced iterative reducing and clustering using hierarchies (BIRCH), clustering using representatives (CURE), robust clustering using links (ROCK)...
    - Diversive
      Diversive analysis (DIANA), monothetic analysis (MONA) ...
- Centroid-based clustering
  Centroid-based clustering represents clusters by a central vector.
    - K-means, iterative self-organizing data analysis technique (ISODATA), generic K-means algorithm (GKA), partitioning around medoids (PAM), fuzzy c-means (FCM)...
- Distribution-based clustering
  Distribution-based clustering defines clusters as objects belonging most likely to the same distribution. This closely resembles sampling random objects from a distribution which is how artificial data sets are generated.
    - Gaussian mixture density decomposition (GMDD), AutoClass ...


• Density-based clustering
  Density-based clustering defines clusters as areas of higher density than the
  remainder of the data set. Sparse areas separate clusters and are
  considered to be noise and border points.
  - DBSCAN, OPTICS, CLARA, CURE, CLARANS, BIRCH, DENCLUE, WaveCluster, FC, ART ...

[16][5]

• Flow-based clustering
  Flow-based clustering algorithms are based on the idea that flow between
  nodes in a cluster should be higher than the flow between other nodes in
  the graph.[8]

The chosen clustering algorithm to be used in this thesis is a flow-based clustering algorithm based on minimum cuts developed by Flake et al. The motivation for choosing this algorithm is that Flake et al. have used it in studies similar to the one chosen for this thesis but instead of focusing on collaborations it investigated citation networks in scientific literature. One aim for this study is to evaluate whether the Flake algorithm is applicable for the chosen problem.

2.2 Earlier work

2.2.1 Clustering based on citations

There seems to be a fair amount of previous work where the clustering of scientific literature is based on citations. Thus, like the experimental study of Flake et al. representing the documents as nodes and the citations as edges. For example L. Leydesdorff [28] where the focus is on finding bi-components and their sizes. Another example is B. Aljaber et al. [9] where documents are clustered based on the utilization of citation contexts. Also L. Bolelli et al. [15] who characterizes Document clustering algorithms as text-based [18][19][20], link-based [31][13] and hybrid.[10][30][14].

2.2.2 Studies by Flake et al.

Flake et al. [12] has conducted experimental studies using their algorithm to partition the data from three different sources. Two experiments were the Open Directory Project and the 9/11 community, where web pages were represented as nodes and the hyperlinks were represented as edges.

Another experimental study was applying the algorithm to the digital library for scientific literature, CiteSeer. In this study the documents were represented as nodes and the citations between documents were represented as edges. This is a fairly similar application to ours except that the authors are not represented in the graph in any way. Instead the graph is purely a representation
of the documents and their citations. According to Flake et al. the iterative
cut-clustering algorithm yielded a high-quality clustering both on the large and
small scale.

Flake et al. showed as a conclusion that minimum cut trees, based on
expanded graphs, provide means for producing a quality clustering and extract-
ing heavily connected components. This provided the basis for us selecting the
algorithm provided by Flake et al. for our work.

The literature study showed that representation of scientific literature as nodes
and citations between them as edges is a common approach. The literature
study did not however, point towards the representation of researchers as nodes
and collaborations as edges being common. Furthermore it is certain that no
such work has earlier been conducted at KTH.

2.3 Research at KTH

KTH has designated five areas as focus areas of research. These focus areas are
derived from challenges which KTH considers to be some of humanity’s great-
est challenges. The five focus areas are Transport, Life Science Technology,
Materials, Information and Communication Technology and Energy [26]. The
research is not however contained by these five focus areas but rather organised
into schools which in turn incorporates research departments. The schools are:
Architecture, Built Environment, Biotechnology, Chemical Science and Engi-
neering, Computer Science and Communication, Education and Communication
in Engineering Science, Electrical Engineering, Engineering Science, Industrial
Engineering and Management, Information and Communication Technology and
Technology and Health. These schools are then divided into departments, cen-
tres and bachelor programs [3]. This thesis will only study the schools and
departments.

2.4 Graph Theory

The chosen algorithm developed by Flake et al. is a flow-based algorithm find-
ing maximum flows in graphs. Therefor graph theory and the theory of flow
networks has been studied.

2.4.1 Flow Network

A flow network is a directed graph $G = (V, E)$ where $V$ is the node set and $E$ is
the edge set. $G$ has two special nodes $s$ and $t$, $s$ is the source and $t$ is the sink.
Each edge has a non negative capacity function $c(v, w)$. For each edge $(v, w)$
the edge $(w, v)$ is also present. A flow $f$ on $G$ is a real-valued function on the
edges that satisfies the following constraints [11]:

$$\begin{align*}
\text{Flow conservation:} & \sum_{(u, v) \in E} f(u, v) - \sum_{(v, u) \in E} f(v, u) = 0, \\
\text{Capacity constraint:} & f(v, w) \leq c(v, w) \text{ for all } (v, w) \in E.
\end{align*}$$
\[ f(v, w) \leq c(v, w) \text{ for all } (v, w) \in V \times V. \text{ (Capacity constraint)} \]

\[ f(v, w) = -f(w, v) \text{ for all } (v, w) \in V \times V. \text{ (Antisymmetry constraint)} \]

\[ \sum_{u \in V} f(u, v) = 0 \text{ for all } v \in V - [s, t]. \text{ (Flow conservation constraint)} \]

Ford and Fulkerson only wrote about the flow between the source and the sink, however W. Mayeda [29] introduced the multi-terminal problem where flows are considered between all pairs of nodes in a network and R. T. Chien [7] discussed the synthesis in such a network.[24]

### 2.4.2 Karzanov’s Preflow

A preflow [27] is similar to a flow except that the total amount flowing into a vertex can exceed the total amount flowing out. Karaznov’s algorithm maintains during each phase a preflow in an acyclic network. The algorithm pushes flow through the network to find a blocking flow, which determines the acyclic network for the next phase.[22]

### 2.4.3 Minimum cut, maximum flow

A cut in G is defined as partitioning V into two subsets A and B, such that:

\[ A \cup B = V \text{ and } A \cap B = \emptyset. \]

The cut is denoted as \((A, B)\). The sum of the weights of the edges crossing the cut defines the value of the cut. The real function \(c\) represents the value of a cut. So, cut \((A, B)\) has value \(c(A, B)\).

The goal of the minimum cut problem is to find the cut that has the minimum cost amongst all possible cuts on a graph. The minimum cut between two nodes \(s, t\) in G can be found by inspecting the path that connects \(s\) and \(t\) in \(TG\). The edge with minimum capacity on that path corresponds to the minimum cut. The capacity of this edge is equal to the minimum cut value, and its removal yields two sets of nodes in TG, but also in G, which correspond to the two sides of the cut.

The minimum cut can also be found by finding a maximum flow from the source \(s\) to the sink \(t\). A maximum flow from \(s\) to \(t\) saturates a set of edges in the graph dividing the nodes into two disjoint parts corresponding to a minimum cut. Therefore the minimum cut and maximum flow problems are equivalent.[21]
3 Methods

As earlier mentioned the goal of this thesis has been to investigate the possibility of using a clustering algorithm to find a structure of collaboration groups, which is simply defined as researchers connected through collaborations between each other. The studied university has been KTH.

A web crawler was used to retrieve data from the website of the publication database DiVA. Two researchers being co-authors on a research publication defined a collaboration between the two.

The aim was to use a flow based clustering algorithm developed by Flake et al. [12] that uses minimum cut trees to find clusters and see if it yielded a result that could create a beneficial collaboration group structure.

3.1 DiVA

DiVA is an online research publication database. All publications written by researchers in their capacity of KTH employees should be registered in DiVA accordance with the KTH president’s decision V-2010-0482 [1]. Each researcher has therefore since 2006 been responsible for registering their own publications in DiVA.

3.2 Web crawler

The web crawler tool Scrapy was chosen to collect the information from DiVA since it is easy to customise in python and has extensive documentation and has a large open source community. The crawled information was sent through a Scrapy pipeline to a Mysql database.

The crawl on DiVA had the following constraints: The article must have been published in the years 2015 or 2016 and must be an article published in a journal and refereed. These constraints were chosen to limit the amount of collected data.

From the articles that met these constraints every author at KTH with a person ID were collected and if two or more of these authors were co-authors on an article there was considered to be a collaboration between all the them. For each author name, person ID, school and department at KTH was collected.

The person ID was not readily obtainable from any page but was however used in links and could therefore be extracted using regex.

All school and department information was contained in one string, where sometimes previous employments existed as well. The beginning of the string was assumed to consist of the current or most recent employment so the school and department information was collected according to that assumption. Information about the author was only collected if the most recent employment was at KTH.

Researchers who lacked school or department information at KTH at their
most recent employment in DiVA were sent to the database with an empty string as school and/or department.

### 3.3 Database

The database consisted of two tables: Edge(id, source, target, weight) and Node(id, name, school, department, collaboration_group).

A node represented a researcher and each researcher had a unique person ID collected from DiVA. School and department at KTH were also stored in the table. The collaboration group column had the initial value null and was later filled with the collaboration group number given to some researchers by the implemented cut-clustering algorithm.

All combinations of the authors of a report were inserted into the database as edges where the source and target consisted of their person ID:s. Source and target in the Edge table were therefore foreign keys of person ID in the Node table. First, all collaborations were inserted into the Edge table without weight. At insert time each node was given a unique ID, which was the primary key.

Researchers from centers, superseded departments, closed departments and researchers who lacked school and/or department information were removed from both the Edge and Node table using SQL-queries.

Using Java JDBC, all collaborations between two authors were added and the weight of the edge between two researchers was set to the number of collaborations between those two researchers. The assigning source and target to specific authors was arbitrary since the collaborations were two-way and the graph undirected. Furthermore a collaboration was defined as two researchers that has collaborated at at least one journal article.

### 3.4 Graph clustering algorithm

The algorithm developed by Flake et al. [12] is based on maximum flow techniques and in particular min-cut trees. By inserting an artificial sink into a graph and connecting it to all nodes via an undirected edge of capacity \( \alpha \), maximum flow can be calculated between all nodes of the network and the artificial sink.

To calculate the min-cut tree the Gusfield variant of the Gomory-Hu algorithm was chosen for this study. To find each minimum cut in the Gusfield Gomory-Hu algorithm the Push-relabel algorithm was chosen.

After finding the Gomory-Hu tree the artificial sink is removed leaving separate components as clusters as a result.

\[
\text{CutClustering\_Algorithm}(G(V, E), \alpha) \\
\text{Let } V' = V \setminus t
\]
For all nodes \( v \in V \)

Connect \( t \) to \( v \) with edge of weight \( \alpha \)

Let \( G'(V', E') \) be the expanded graph after connecting \( t \) to \( V \).

Calculate the min-cut tree \( T' \) of \( G' \)

Remove \( t \) from \( T' \)

Return all connected components as the clusters of \( G \)

3.4.1 Gomory-Hu


The Gomory-Hu algorithm is based on the property that if \((S, V \setminus S)\) is a minimum \( s \)-\( t \)-cut in \( G \) and \( u,v \) are any pair of vertices in \( S \) then there exists a min-\( s \)-\( t \)-cut \((S*, V \setminus S*)\) such that \( S*S \).

The algorithm works recursively and distinguishes between original and contracted nodes.

An original node is a vertex of the input graph. The algorithm picks two original nodes, \( s \) and \( t \), if there are more than one original node. A minimum \( s \)-\( t \) cut \((S,T)\) is then found, and two graphs \( G_s \) and \( G_t \) are formed. \( G_s \) is formed by contracting \( S \) into one node, and \( G_t \) is formed by contracting \( T \). The algorithm then continues to build cut trees in \( G_s \) and \( G_t \). The capacity of each cut is the sum of the weights crossing the cut. A trivial cut around a contracted node is a minimum cut in the graph.

When there is only one original node the algorithm has enough information to construct a cut tree and the recursion bottoms out. [23] [17]

3.4.2 Gusfield

The Gusfield algorithm finds min-cut trees but avoids contraction operations. Gusfield essentially provides rules for adjusting min-\( u \)-\( v \)-cuts that potentially cross. These cuts were found iteratively in \( G \) (the original graph) instead of in \( G_s \).

The Gusfield algorithm iteratively calculates the minimum cut. At each iteration a new vertex is chosen as the source. [25]

3.4.3 Push-relabel

The excess \( ef(v) \) is defined as the difference between the incoming and the outgoing flows. If a flow on an edge can be increased without violating the capacity then that edge is residual, otherwise it is saturated. The residual capacity \( uf(v, w) \) of an edge \((v, w)\) is the amount by which the edge flow can be increased. The
residual edges make up the residual graph.

If preflow is notated as f, initially f is equal to zero on all edges and the excess is zero on all nodes except s. The excess on s is set to a number that exceeds the potential flow value.

Distance labeling \( d : V \rightarrow N \) satisfies the following conditions:

- \( d(t) = 0 \)
- For every residual edge \((v, w)\), \( d(v) \leq d(w) + 1 \)
- A residual edge \((v, w)\) is admissible if \( d(v) = d(w) + 1 \)
- A node \( v \) is active if \( v \notin s, t, d(v) \parallel n, \) and \( e_f(v) > 0 \)

Initially \( d(v) \) is the smaller of \( n \) and the distance from \( v \) to \( t \) in \( G_f \). The method repeatedly performs update operations: push and relabel. The pushing operation updates the preflow and the relabeling operation updates the distance labeling. Pushing from \( v \) to \( w \) increases \( f(v, w) \) and \( e_f(w) \) by \( \delta = \min\{e_f(v), u_f(v, w)\} \), and decreases \( f(w, v) \) and \( e_f(v) \) by the same amount. Relabeling \( v \) sets the label of \( v \) equal to the largest value allowed by the valid labeling constraints.

**push\((v, w)\)**

**Applicability:** \( v \) is active and \((v, w)\) is admissible.

**Action:** send \( \delta = \min\{e_f(v), u_f(v, w)\} \) units of flow from \( v \) to \( w \).

**relabel\((v)\)**

**Applicability:** \( v \) is active and push\((v, w)\) does not apply for any \( w \).

**Action:** replace \( d(v) \) by \( \min_{(v, w) \in E_f} d(w) + 1 \),

or by \( n \) if \( \neg \exists (v, w) \in E_f \).

The order in which these operations are performed determines how efficient the method will be.

An edge of \( G \) is an unordered pair \( \{v, w\} \) such that \((v, w) \in E\).

The three values \( u(w, v), u(v, w) \) and \( f(v, w)(= - f(w, v)) \) are associated with each edge \( \{v, w\} \). Each node \( v \) has a list of incident edges \( \{v, w\} \), in fixed but arbitrary order. Thus each edge \( \{v, w\} \) appears in exactly two lists, the one for \( v \) and the one for \( w \). The current edge \( \{v, w\} \) of a node is the current candidate for a pushing operation from \( v \). The current edge is initially the first edge on the edge list of \( v \).
The push-relabel method consists of one main loop repeating the discharge operation until there are no active nodes.

discharge(v)

**Applicability:**  
v is active.

**Action:** let \(\{v, w\}\) be the current edge of \(v\);

\[
\text{end-of-list} \leftarrow \text{false};
\]

repeat

if \((v, w)\) is admissible then push\((v, w)\)

else

if \(\{v, w\}\) is not the last edge on the edge list of \(v\) then

replace \(\{v, w\}\) as the current edge of \(v\) by the next edge on the list

else begin

make the first edge on the edge list of \(v\) the current edge;

\text{end-of-list} \leftarrow \text{true};

end;

until \(e_f(v) = 0\) or end-of-list;

if end-of-list then relabel\((v)\);

The discharge operation iteratively checks if a pushing is admissible then pushes the excess at \(v\) through the current edge \(v, w\) of \(v\). If pushing is not admissible the the operation replaces \(v, w\) as the current edge of \(v\) with the next edge on the list of \(v\). If \(v, w\) is the last edge on this list, it makes the first edge on the list the current one and relabels \(v\). When the excess at \(v\) is reduced to zero the loop will end.

When there are no active nodes the first stage of the push-relabel method terminates, the excess at the sin is equal to the minimum cut value and the set of nodes which can reach the sink in \(G_f\) induces a minimum cut.

The second stage of the method converts \(f\) into a flow \([6]\).
3.5 Implementation of algorithm

3.5.1 Graph data structure

By using Java JDBC the node and edge information was requested from the database and then put into a data structure that the algorithm could operate on. The data structure consisted of the three classes Node, Edge and Graph.

The Node class contained person ID as a string and a unique integer ID. Each node ID in a Graph was unique and mapped to a specific person ID. The Edge class contained a unique integer ID, two Node objects and weight. The Graph class contained three Java HashMaps. The first HashMap mapped each node to it’s ID, the second HashMap mapped each edge to it’s ID and the third HashMap mapped each node to it’s person ID. To operate on the graph a method was implemented to obtain the graph’s adjacency matrix.

3.5.2 Extracting components

Since the Flake et al. algorithm only operates on a connected graph all the connected components of the graph needed to be obtained [12]. This was done by implementing a breadth-first based algorithm going through the whole graph finding all connected components.

The connected components were constructed into separate Graph objects and then put in a data structure called ComponentContainer containing all components in a list. The ComponentContainer enabled the retrieving of information about mean size of components, fetching all components of a specified size etc.

3.5.3 Choosing $\alpha$

For the Flake et al. algorithm to yield the desired result an appropriate value for $\alpha$ is needed, see 3.4. This was done by running the cut-clustering algorithm several times on the graph with $\alpha$-values increasing from 1. As noted by Flake et al. [12] the clustering goes towards one large component as alpha goes to 0 and towards all nodes as single components as $\alpha$ goes to infinity.

The goal was to place as many researchers as possible in groups of a desired size and an upper and a lower limit was chosen for group sizes. The idea was to find the $\alpha$-value that yielded the highest quantity of components larger than the lower limit and the lowest quantity of single node components.

So for each $\alpha$-value the number of single node components and the number of returned components larger than the lower limit were retrieved. These two numbers were then subtracted and stored in a list.

The algorithm finished after finding a partitioning with only single node components. Since the goal was to minimise the number of single node components and maximise the number of large components the components from
the partitioning with an $\alpha$-value that yielded the smallest subtraction of single node components and large components were returned.

### 3.5.4 Choosing group sizes

The group sizes in this study was limited to groups from 6 to 20 researchers. There is no consensus of the optimal groups size but there are studies that shows that groups of five people or smaller as are considered too small and groups of sizes 12 or larger are considered too big [2].

The idea in this study was to create groups with a mean size of 6-12 people in each group and therefore the limit 6-20 was chosen.

### 3.5.5 Returning clusters

Because the research graph created from the DiVA-data consisted of several components, all connected components of the graph were found and put in a ComponentContainer, holding connected components. First, the components that were already of the desired collaboration group size were put in a new ComponentContainer that were to hold all retrieved collaboration groups. All single node components (researchers who had not collaborated with anyone else at KTH) were disregarded. The algorithm then processed all components larger than 20 nodes.

The procedure of collecting components of desired size, disregarding single node components and running the algorithm on the components larger than 20 were done recursively.

Finally clusters of desired size were returned and considered as collaboration groups of researchers. The Node table in the database were then updated with integer values representing the collaboration group each researcher was found in after running the algorithm. Researchers who did not collaborate with anyone to begin with or who was not placed by the algorithm in a collaboration group with other researchers kept the value null as their collaboration group.

### 4 Result

#### 4.1 Collected data from DiVA

From the data that was collected by using the web crawler with specific constraints, see 3.2, it was found that 0.6% of the collected authors lacked school information and 2.6% lacked department information. These were disregarded and removed from the database. After the removal of these researchers 2261 researchers remained. Between these researchers 3467 collaborations occurred, for definition of collaboration see 3.3.
Of the 3467 found collaborations 8% were found to be cross school collaborations and 15% cross department collaborations.  
16% of the collected researchers had not collaborated with anyone at KTH during the years of 2015 & 2016 and were subsequently excluded.  
The weights in the graph were for the most part very low, most researchers only collaborated with another researcher 1 time which can be seen in Figure 1.

Figure 1: Size of weights in graph

Figure 1 displays the proportions of different weight sizes in the graph. As can be seen 71% of the weight were of size 1 and 93% were of size 4 or less.

4.2 Collaboration groups

When the running the algorithm the constraint of groups of size 6-20 researchers were chosen, as mentioned earlier. Figure 2 displays the distribution of all researchers in the database.
As can be seen in Figure 2 only 34% of the total number of researchers from the database were actually put in a collaboration group. And as stated in 4.1 16% were automatically disregarded because there were no edges connected to them. 50% of researchers were not placed in any collaboration groups by the algorithm even though they had collaborated with at least one other researcher.

In total 77 collaboration groups were created and had a mean size of 10 researchers per group. Of those groups 23% consisted of researchers from several schools at KTH, and 41% from several departments.
Figure 3 and 4 shows how many percentage of the groups that contained researchers from 1, 2 or 3 different schools and departments respectively. No groups contained more than 3 schools or departments. What can also be seen in Figure 3 and 4 is that there are more collaboration groups with people collaborating between departments than between schools.
4.2.1 School representation in collaboration groups

Figure 5: Number of researchers from each school among all researchers in the database

Figure 6: Number of researchers from each school among the researchers in all collaboration groups
Figure 5 and 6 shows the different schools represented among all the researchers in the database and the representation among the researchers placed in a collaboration group.

It can be noted that the School of Education and Communication in Engineering Science (ECE) are not displayed in Figure 5 and 6. The reason is that only 12 researchers from ECE exist in the original database and none of them are placed in collaboration groups and were therefore disregarded.

4.2.2 Collaborations per researcher

A standard was proposed to measure to which extent a collaboration group is connected. This standard was defined as number of collaborations per researcher in a group. It was calculated by adding all the weights of all the collaborations between all the researchers in the group and then dividing this sum by the number of researchers in the group. All values of collaborations per researchers for all collaboration groups are displayed in Figure 7.

What can be noted in Figure 7 is that collaboration group number 76 has much higher value of collaboration groups per researcher. Group number 76 has a value of 133 collaborations per researchers while the value for all other groups varies from 1-8 which can be found in Figure 8.
By looking in the database it was found that all researchers in collaboration group number 76 works at the Physics department at the School of Engineering and Sciences, see Figure 9.

### Figure 9: Collaboration group number 76

<table>
<thead>
<tr>
<th>Name</th>
<th>School</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lund-Jensen, Bengt</td>
<td>School of Engineering Sciences (SCI)</td>
<td>Physics</td>
</tr>
<tr>
<td>Kuwertz, Emma</td>
<td>School of Engineering Sciences (SCI)</td>
<td>Physics</td>
</tr>
<tr>
<td>Morley, Anthony</td>
<td>School of Engineering Sciences (SCI)</td>
<td>Physics</td>
</tr>
<tr>
<td>Sidebo, Edvin</td>
<td>School of Engineering Sciences (SCI)</td>
<td>Physics</td>
</tr>
<tr>
<td>Strandberg, Jonas</td>
<td>School of Engineering Sciences (SCI)</td>
<td>Physics</td>
</tr>
<tr>
<td>Jovicevic, Jelena</td>
<td>School of Engineering Sciences (SCI)</td>
<td>Physics</td>
</tr>
</tbody>
</table>

By retrieving the collaborations with highest weight from the database it was also found that the 12 collaborations with highest weights were between researchers from group number 76, see Figure 10.
Figure 10: Collaborations with highest weight

<table>
<thead>
<tr>
<th>Researcher 1</th>
<th>Researcher 2</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strandberg, Jonas</td>
<td>Lund-Jensen, Bengt</td>
<td>116</td>
</tr>
<tr>
<td>Strandberg, Jonas</td>
<td>Morley, Anthony</td>
<td>91</td>
</tr>
<tr>
<td>Morley, Anthony</td>
<td>Lund-Jensen, Bengt</td>
<td>89</td>
</tr>
<tr>
<td>Lund-Jensen, Bengt</td>
<td>Kuwertz, Emma</td>
<td>68</td>
</tr>
<tr>
<td>Morley, Anthony</td>
<td>Kuwertz, Emma</td>
<td>68</td>
</tr>
<tr>
<td>Strandberg, Jonas</td>
<td>Kuwertz, Emma</td>
<td>67</td>
</tr>
<tr>
<td>Kuwertz, Emma</td>
<td>Jovicevic, Jelena</td>
<td>60</td>
</tr>
<tr>
<td>Lund-Jensen, Bengt</td>
<td>Jovicevic, Jelena</td>
<td>60</td>
</tr>
<tr>
<td>Morley, Anthony</td>
<td>Jovicevic, Jelena</td>
<td>60</td>
</tr>
<tr>
<td>Strandberg, Jonas</td>
<td>Jovicevic, Jelena</td>
<td>59</td>
</tr>
<tr>
<td>Sidebo, Edvin</td>
<td>Lund-Jensen, Bengt</td>
<td>29</td>
</tr>
<tr>
<td>Strandberg, Jonas</td>
<td>Sidebo, Edvin</td>
<td>29</td>
</tr>
</tbody>
</table>

5 Discussion

5.1 Interdisciplinarity in collected data

In the results some interesting tendencies can be observed. First of all it seems as though the interdisciplinary collaborations are still not so widespread at KTH. Looking at the data retrieved directly from DiVA only 8% of the total number of collaborations were between two different schools. Even the number of cross department collaborations can be considered to be few since it was only 14% of the total number of collaborations. This might suggest that most researchers still essentially collaborate with researchers in their own field.

The number of unique schools and departments in the retrieved collaboration groups were also mostly 1 and in some cases 2, or 3, as displayed in Figure 3 and 4. This indicates that the collaborations are still mostly between researchers with similar academic background.

5.2 Evaluation of collaboration groups

77 collaboration groups have been suggested by using the chosen cut-clustering algorithm developed by Flake et al. Even though most of them are not cross department they still locate groups of researchers currently collaborating with each other. These groups could therefore hopefully be beneficial for the affected researchers, especially if interdisciplinary research over time become more popular.

To evaluate how accurate these collaboration groups are it is interesting to look at the school representation among the researchers before and after running
the algorithm. Figure 5 and 6 showed that the ratio of the school representation was preserved even though not all researchers were placed in collaboration groups. This is an indication that these collaboration groups reflect the actual relationships within and between the schools.

The collaboration groups were also of desired size since the goal was to find groups with a mean size of 6-12 researchers per group, see 3.5.4, and the result was 10.

Furthermore a standard of measuring the connectedness of the collaboration groups was suggested and defined as collaborations per researcher in each group. The main idea behind this is to enable future research groups of this kind to be easily compared. Another outcome of comparing collaborations per researcher among the suggested groups were that one group was found to have an especially high connectedness. In Figure 7 it can be seen that collaboration group number 76 has a profoundly higher rate of collaborations per researcher than for the rest of the groups.

By querying the database it could be shown that these authors were all from the Physics department at the School of Engineering Sciences and that the weights of their collaborations were the highest in the entire database. The highest weight was as much as 116, which means that these researchers should have been co-authors on 116 reports during 2015 and 2016. Since 93% of the collaborations as showed in Figure 1 only had a weight of 4 and less this might reveal an inconsistency in the collected data and an inconsistency in the routines for submitting report information to DiVA.

5.3 Limitations of DiVA

The DiVA database have been shown to be in many ways inconsistent and lacking information. For example, information about department and/or school affiliation were missing for some researchers, others lacked person ID. There was neither any information about what the system for presenting current and previous employment is. In this study the first employment in the string collected by the web crawler was considered to be the most recent which was only based on looking up some of the authors’ web pages where their employment history could be found.

Another problem was revealed in previous section where some authors at the Physics department appeared to have been writing as many as 116 reports together during the past two years. This reveals that there is no decree for which researchers should be found in the author list for a report.

These problems are probably partially due to the fact that the researchers themselves are responsible for registering their own publications. If the registration would have been more directed these inconsistencies would hopefully have been fewer. An improvement of the DiVA website or at least access to their actual database could be beneficial in further studies.
5.4 Further studies and possible improvements

A first approach for future research would be to run the algorithm with different group size specification and with different amount of data. For example several types of articles could be considered and not simply journal articles. Also the time interval could be expanded to years beyond 2015 and 2016. Thereafter these results could be compared with the results from this work by looking at collaborations per researcher and looking at how many researchers were placed in groups. Both of these measurements can be an indication of how appropriate the collected groups are. A poor result would be if most of the researchers were disregarded by the algorithm or if the rate of collaborations per researcher were very low. Therefore both of these aspects should be taken into account.

The results from this work were not optimal since 50% of the researchers were not placed in collaboration groups. One theory of why this was the case is that the graph created was not weighted enough. As stated by Flake et al. the cut-clustering algorithm performs best on weighted graphs.[12] And as can be seen in Figure 1 71% of the weights are of weight 1.

It is also worth noting that only researchers that have collaborated with another researcher during the years of 2015 and 2016 are considered for collaboration groups. 16% of the found researchers are therefore disregarded immediately. Also, the researchers that had not published any work in 2015 or 2016 were not even placed in the database and therefore also disregarded. Depending on the desired result, it could be problematic that these people will not be placed into collaboration groups. The goal of this project however has not been to place every researcher into collaboration groups, only to locate clusters.

As stated earlier several authors could be reported as co-authors of an article, however DiVA has no available information on how much each author has actually contributed. Therefore all contributions are considered equal. If a measurement was provided from DiVA stating how much each co-author has contributed, the weights of the graph could be adjusted to mirror this.

A possible approach could be to develop methods of deciding how strong a collaboration is. Basically collaboration weight could not only be decided by looking at how many reports two researchers have co-authored. A suggestion would be to develop a measurement of the involvement of a researcher in an article. However difficulties may arise when attempting to consider the aspects necessary to gain a result that is close to reality.

Citation could be one an interesting approach when deciding the weight of the collaborations. If a report has been cited many times that collaboration could be considered as stronger. This might yield some interesting results.

Another interesting future approach could be to try other clustering algorithms and compare the results. For example it could be appropriate to test other types of clustering algorithms as those stated in section 2.1.

Finally it would be interesting to proceed by applying the suggested method on other universities where the interdisciplinary research might be more widespread than at KTH and compare the results to the ones in this thesis.
6 Conclusion

From these results it seems as if researchers mostly still collaborate within their own department and that interdisciplinary research is not that widespread at KTH yet. Still almost half of the 77 suggested collaboration groups consisted of researchers from several departments. Especially if the interdisciplinary work increase at KTH this method for finding collaboration groups could be beneficial for the researchers and for KTH in general.

Even though not all researchers were placed in collaboration groups the results still showed that clusters of researchers collaborating with each other can be found using the algorithm developed by Flake et al. The school representation among all researchers and among those placed in collaboration groups were contained after running the algorithm and the groups were of desired size according to the specifications made. By increasing the amount of data and further investigating how a collaboration can be accurately defined, a graph more suitable for the Flake algorithm can hopefully be gained to further enhance the results.

Also, problems with inconsistencies in how researchers are stated as authors were revealed by introducing a measurement of connectedness in the found collaboration groups.

Finally further research with different definitions of a collaboration and other clustering algorithms is encouraged. Future work could hopefully suggest a beneficial structure of collaboration groups that is not based on academic background but on actual research collaborations. This could possibly be beneficial not only for KTH but for other universities as well, especially the ones where the interdisciplinary research is more widespread.
References


