BADM500

(10 ECTS - Spring Semester)

Bachelor’s Project in Computer Science

Bachelorprojekt i datalogi

This booklet describes the procedure to find an advisor and a topic for the Bachelor projects in Computer Science in the Spring of 2021. In particular, it contains a catalogue of proposals for project topics prepared by the available advisors.

Overview

Your advisor and project topic will be decided in one of the following two ways:

- Private agreement with the advisor on topics not in the catalogue
- Central assignment system for the topics in the catalogue

You are free to choose which of the two ways you want to use. However, if you make private agreements you still must provide your input to the central system as this information will be used to distribute equally the workload among advisors. Your private agreement will be communicated to the central system by the advisor.

By ministerial regulations Bachelor projects are to be carried out individually or in groups of at most four persons. Groups are in all cases formed by the students themselves, neither the advisor nor the central system will enforce groups, if they have not been decided by mutual agreement by the students.

The exam consists of a written report. In addition, students that work in groups will have to undergo an oral exam to assess the individual performance. Oral exams in case of students that conducted individual projects are not needed by regulation and will be up to the choice of the advisor.

Timeline

- November 20-30: registrations to BADM500 via SPOC
- November 30 deadline for making private agreements.
- December 2: information meeting via asynchronous chat in the BADM500 2021 Community in MS Teams
- December 2-6 (at noon): submission of preferences about the topics in the catalogue at https://valkyrien.imada.sdu.dk/BADM500
- December 9-13: assignment of students to project topics
- December 15 ca. Publication of the assignment.

To join the BADM500 2021 Community in Microsoft Teams:

1. Click the ‘Teams’ button on the left side of the app, then click ‘Join or create a team’ at the bottom of your teams list.
2. Go to ‘Join a team with a code’ (the second tile), paste the code **dco2o2p** in the ‘Enter code’ box, and click ‘Join’.

Submission of Preferences

Preferences are expressed by a ranked list of 5 topics among those offered in the catalogue. Ties are not allowed. For self-formed groups, each member of the group must submit the usernames of the other members of the group and exact same preferences as them. Note that in your list you cannot rank a topic with a maximum group size lower than the size of your group. On the other hand, there might be topics where the advisor does not accept students working individually - check the preferences for the group size of each topic.

Central Assignment System

The central assignment system is an algorithm that tries to maximize the total satisfaction based on the preferences submitted (that is, it tries to assign as many as possible students to their best preferences) subject to the capacity restrictions and the achievement of a form of fairness consisting in giving the best possible treatment to the worst off student.
A more detailed description of the algorithm used for the assignment is available in the Appendix of this document.

Topic ID: 1
Advisor: Joan Boyar
Project title: Exact algorithms for the bounded degree vertex cover problem
Group Size: min 1 and max 4 students
Project description:
Vertex Cover is a well known NP-hard problem, and it is even NP-Complete on graphs with maximum degree at most 3 (bounded degree). The problem is to find a minimum size subset C of the vertices of a given graph G such that every edge in G has at least one endpoint in C.

The Vertex Cover problem has several applications. For a straight-forward example, consider a computer network represented as a graph, with the routing servers being the vertices and the connections between them being the edges. To prevent the propagation of worms in the network, having a defense at those router servers forming a minimum vertex cover is most efficient.

An exact algorithm solves the problem (it does not only find an approximate solution). Of course, no polynomial time algorithms are known. However, there are algorithms with running time $O(q^n(poly(n)))$, where there are n vertices, $q<2$, and poly(n) is a polynomial in the number of vertices.

The goal of this project is to consider exact algorithms for Vertex Cover when no vertex has more than d neighbors (for example d=3). One possibility is to implement, test, and compare known algorithms. However, there are possibilities for concentrating on the analysis of the algorithms or for designing new algorithms.

Topic ID: 2
Advisor: Lene Favrholdt
Project title: Exact algorithms for the bounded degree vertex cover problem
Group Size: min 1 and max 4 students
Project description:
Vertex Cover is a well known NP-hard problem, and it is even NP-Complete on graphs with maximum degree at most 3 (bounded degree). The problem is to find a minimum size subset C of the vertices of a given graph G such that every edge in G has at least one endpoint in C.

The Vertex Cover problem has several applications. For a straight-forward example, consider a computer network represented as a graph, with the routing servers being the vertices and the connections between them being the edges. To prevent the propagation of worms in the network, having a defense at those router servers forming a minimum vertex cover is most efficient.

An exact algorithm solves the problem (it does not only find an approximate solution). Of course, no polynomial time algorithms are known. However, there are algorithms with running time $O(q^n(poly(n)))$, where there are n vertices, $q<2$, and poly(n) is a polynomial in the number of vertices.

The goal of this project is to consider exact algorithms for Vertex Cover when no vertex has more than d neighbors (for example d=3). One possibility is to implement, test, and compare known algorithms. However, there are possibilities for concentrating on the analysis of the algorithms or for designing new algorithms.

Topic ID: 3
Advisor: Kim Skak Larsen
Project title: Advanced Topics in Compiler Construction
Group Size: min 2 and max 4 students
Project description:
A modern compiler is organized into phases, where the basic ones cover lexical and syntactic analysis, resulting in an abstract syntax tree (AST). Subsequent phases adorn and analyze the AST, building a symbol table, performing type checking, and finally generating target code (assembler, for instance). This material is known from DM565 for a simple imperative language.

The goal of this project is to go beyond this basic language and basic compiler. This would include extending the language with more types, classes, or other features. It would also include extending the compiler with advanced optimization techniques such as liveness analysis and register allocation, peep-hole optimization, garbage collection, etc. One could possibly focus on some domain-specific language rather than a general-purpose programming language.

The project will be evaluated on language design and usability, extent and effectiveness of (optimization) phases, and maintainability. For the connection between the implementation and the report, the evaluation will focus on motivation, descriptions of advanced features relative to a basic language and compiler, and documented correctness and effect.

Topic ID: 4
Advisor: Fabrizio Montesi
Project title: Implementation and optimisation of function calls in Jolie
Group Size: min 1 and max 3 students
Project description:
The goal of this project is to extend the interpreter of the Jolie programming language (https://jolie-lang.org) to support calling external services (e.g., web services using HTTP) as if they were local functions inside of a program. This requires understanding the relevant components of a real-world interpreter, which is implemented in Java, designing the extension, implementing it, and documenting it. In this process, if time allows, there is also space for improving on the implementation of service calls, by developing an optimisation for copyless message passing: a more efficient implementation of message sending where the content of the message is not copied from memory but rather passed by reference, which requires a static analysis to check whether the optimisation is safe. The project includes learning how to prepare an official pull request for inclusion into an open source project (https://github.com/jolie/jolie), following its guidelines for documentation and testing.

Topic ID: 5
Advisor: Luís Cruz-Filipe
Project title: Sorting networks
Group Size: min 1 and max 2 students
Project description:
Sorting networks are oblivious algorithms that sort a fixed number of inputs by applying a predetermined sequence of compare-and-swap operations. These operations may sometimes be applied simultaneously, giving two measures for the complexity of a sorting network: size (number of compare-and-swap operations) and execution time (number of computation steps).

In spite of their simplicity, determining optimal values for the size and execution time of a sorting network (the minimal number of compare-and-swap operations) in terms of the length n of the input is an extremely complex problem that has only been solved for small values of n. The goal of this project is to approach this problem from one of two directions:
- exploring heuristics for generating more efficient sorting networks than those currently known, in order to improve upper bounds for these values;
- looking into the theory of sorting networks in order to obtain theoretical results that can help improve currently known lower bounds for these values.

Topic ID: 6
Advisor: Richard Roettger
Project title: Machine learning based stratification of cancer subtypes
Group Size: min 1 and max 3 students
Project description:
Cancer is not only one of the biggest health risks to society with roughly 15-20 million new occurrences per year, but it is also a very complex and heterogeneous disease. Normally, several factors have to play together to cause the abnormal cell growth which constitutes cancer. On top of that, cancer occurs in many different tissue types showing all different aberrations and genetic marks. These are the reasons why treatment for cancer is still massively invasive with often unknown outcomes and progression prognosis. Even a single cancer type, e.g., breast cancer, can have multiple subtypes of cancer which all require different treatment optimization, from mild to invasive radical. The selection of the appropriate treatment is fundamental.

The aim of the project is to train a machine learning pipeline which allows the classification of various types of breast cancer (depending on availability also different cancer types). You will be given a gene expression dataset from several hundred patients and your goal is to separate the various subtypes in a strategic and robust manner. Since genetic data is very high-dimensional and classically only a few hundred patient samples are available, your project also entails the implementation of suitable feature selection methods and permutation tests to prove the robustness of the implemented classifier. Having robust and interpretable classifiers might help the treatment decision making in the future. The exact scope of the project (depends also on the team size) will be defined in a kick-off meeting where you also get an introduction to gene expression data so you can work with the provided data. Prior knowledge in biology is not required for this project.

Topic ID: 7
Advisor: Jacopo Mauro
Project title: Constraint programming and algorithm portfolio
Group Size: min 2 and max 4 students
Project description:
Constraint programming (https://en.wikipedia.org/wiki/Constraint_programming) is a paradigm for solving combinatorial problems that draws on a wide range of techniques from artificial intelligence, computer science, and operations research. In constraint programming, users declaratively state the constraints on the feasible solutions for a set of decision variables. Constraints differ from the common primitives of imperative programming languages in that they do not specify a step or sequence of steps to execute, but rather the properties of a solution to be found.

One of the most promising approaches to solve constraint problems is to combine the strength of the portfolio of the available solvers to produce a better unique solver called "portfolio solver". In particular, one of the best constraint solvers is sunny-cp (https://github.com/CP-Unibo/sunny-cp) that is using machine learning techniques to select what solvers to run based on the problem to solve. Sunny-cp this year reached the second place in the international constraint competition (https://www.minizinc.org/challenge2020/results2020.html) only after Google Or-Tools solver.

The aims of this project are i) learn how sunny-cp works, ii) improve the internals of sunny-cp and port it to python 3, possibly beating Google's solver in next year’s competition.

Topic ID: 8
Advisors: Stefan Jänicke (main supervisor), Marco Chiarandini, Arthur Zimek
Project title: Visual Data Analysis on Real World Data
Group Size: min 1 and max 2 students
Project description:
Visualization as a means to explore data is especially important if concrete hypotheses on the data at hand are missing or hard to generate, or if state-of-the-art data mining algorithms are hard to apply. In those scenarios, the human capability of perceiving patterns in a visual presentation of the data can lead to gaining insights in the data.
The main focus of a project in this context will be the design of an interactive visual representation of the data to support exploratory visual data analysis, and the application or adaption of state-of-the-art data mining algorithms plays a subordinate role.

Potential projects (might be extended):

- interactive visualization of the Atikamekw territory
- visual alignment of movie subtitles and scripts
- geovisualization for the trade of threatened species in the CITES database
- visualization of timetabling data
- visual historical climate data analysis
- LIVE visualization of natural hazards

Topic ID: 9
Advisors: Arthur Zimek (main supervisor), Marco Chiarandini, Stefan Jänicke
Project title: Data Analysis on Real World Data
Group Size: min 1 and max 2 students
Project description:
In a typical data analysis scenario on real world problems, exploratory data analysis is the first step to learn more about the available data, understand their structure, and test some initial hypotheses. Depending on the gained insights more advanced analysis can be designed and performed.

In this project you will carry out data analysis on one of the following data set arising from real world applications:

- data from rejsekort about the check in and check out of the passengers in public transport in the region of Funen.
- vehicle counters on the urban network from the Odense municipality
- organization of images with geographical information

Other datasets from other contexts might also become available or you might propose your own data set.

The project requires adequate selection, adaptation, and application of data analysis techniques for the tasks of describing the data, predicting patterns and prescribing structural changes. Methods to apply include visualization techniques, feature extraction, cluster analysis, spatial-temporal joins of different data sources, outlier detection, and classification. The ultimate goal is to identify interesting, novel, valid, and potentially useful patterns in (some of) the various datasets; to visualize them; and to prescribe possible improvements in the ways things are currently done.

The project can be cast in several variants with emphasis on various aspects or a combination thereof:

- Analysis and detailed understanding of the data, reporting on potential problems as well as perspectives for further analysis.
- Analysis of the applicability of categories of analysis methods (for example: is it promising to apply clustering methods? (why or why not?); if yes, which kind of clustering is most suitable for the problem?; how can success be evaluated?).
- Identification of advanced approaches from the literature and/or tailoring of methods for the particularities of the available data (e.g., to take into account spatio-temporal relations, interrelations between different data sources, external domain knowledge).

Topic ID: 10
Advisors: Marco Chiarandini (main supervisor), Stefan Jänicke, Arthur Zimek
Project title: Data Science: Optimization
Group Size: min 1 and max 2 students
Project description:
One of the goals of Data Science is to make prescriptions and decisions supported by data. Data-driven decision making tasks can be formulated mathematically as optimization problems, in which decision variables are used to express a desired objective and constraints, often determined by limits on the resources, restrict the values that these variables can take. In this project, we will learn to formalize optimization problems mathematically and in the modelling language MiniZinc. We will then try to solve these problems using different techniques such as constraint programming, integer linear programming and heuristics.
The possible decision making tasks that we will address are:

- instructor assignment at IMADA
- course and exam timetabling at IMADA and at the Faculty of Science
- building maintenance planning at Svendborg Kommune
- ferry routing
- model fitting on COVID data
- sports timetabling https://www.sportscheduling.ugent.be/ITC2021/
- image processing for a hand dexterity test in primary school.

Other similar tasks where decisions have to be taken subject to constraints and optimization goals can be proposed by the students.

Topic ID: 11
Advisors: Jakob L. Andersen, Rolf Fagerberg, Daniel Merkle (primary)
Project title: Parallel Algorithms
Group Size: min 1 and max 4 students
Project description:
Parallel computing can be used to efficiently solve challenging problems in science, industry, and engineering. Supercomputers are required to, e.g., perform complex simulations, to analyze enormous datasets, or, more generally speaking, to perform computations that usually can not be done on a classical desktop or laptop. Worth mentioning, nowadays nearly all computers are parallel because the physics of semiconductor manufacturing will no longer let conventional sequential processors get faster year after year. In order to adapt to this unavoidable situation, *all* efficient implementations need to become parallel programs. Recent standard manycore architectures might easily have more than 60 cores on a single CPU - and even if only one core is used, considering the restrictions of the memory hierarchy of the architecture is crucial for reaching peak performance. The design principles for efficient algorithms on a single core architecture as well as the design principles for efficient parallel algorithms targeted to be executed on thousands of computing nodes of a supercomputer, often follow the same principle: "Use all data with small access time as often as possible".
Therefore, this project is twofold: first, students will work on efficient implementations for classical algorithms like, e.g., matrix matrix multiplication, on a single core. Second, students will work on efficient implementations of parallel algorithms that can scale to millions of computing nodes. The parallel algorithms implemented will also be analysed theoretically wrt. their scalability (using a so-called isoefficiency analysis) and the findings will be compared to the empirical results. Students will very likely have access to Cori, a Cray XC40 supercomputer, which has a peak performance of about 30 petaflops.

Topic ID: 12
Advisors: Jakob L. Andersen (primary), Rolf Fagerberg, Daniel Merkle
Project title: Graph Transformation
Group Size: min 1 and max 3 students
Project description:
In context-free grammars a rule has a single non-terminal on the left-hand side and string on the right-hand side. In context-sensitive grammars the left-hand side is generalized to a string as well. In graph grammars we change from strings to general graphs as the underlying objects. To perform a transformation with a rule L -> R we must then first find L as a subgraph in some input graph, and then try to replace that subgraph with R. In general, this is a difficult problem (NP-hard) and indeed graph grammars are Turing complete, but for some modelling purposes it is natural to use graph transformation.
If we model molecules as graphs we can model chemical reactions with rules. By repeatedly performing graph transformation we can then generate new molecules and the reaction network that connects them.

Depending on the participants this project area may include:

- Implementation of algorithms involved in graph transformation, e.g., subgraph matching, graph isomorphism, the actual transformation, composition of rules.
- Modelling of systems and analysis of results.
- The mathematics and theoretical aspects of graph transformation.

Topic ID: 13
Advisors: Jakob L. Andersen, Rolf Fagerberg (primary), Daniel Merkle
Project title: K shortest paths in graphs and hypergraphs
Group Size: min 1 and max 4 students
Project description:

Finding the shortest path in a graph is a well known problem solved by a number of efficient algorithms taught already in the first year of the Computer Science programme. Can we also find the second shortest path efficiently? Or, more generally, all of the K shortest paths for some chosen value K? This problem has applications, for instance in route finding on road networks (Google Maps often displays alternative routes with only slightly worse length than the optimal path) and in object tracking. Many algorithms have been developed for this problem, including an elegant break-through result by David Eppstein from 1998, which is one candidate for being the focus of this project.

While standard graphs are a good model of road networks and many other phenomena, their generalization to *hypergraphs* is sometimes needed. In hypergraphs, edges are pairs of *sets* of nodes, rather than just pairs of single nodes. Hypergraphs model well collections of chemical reactions in which so-called hyperpaths correspond to synthesis plans for chemical compounds. Thus, finding the K best hyperpaths is a relevant problem when doing chemical synthesis planning in research and industry. Having access to a larger set ($K > 1$) of good paths is important because aspects not captured by the modeling may prevent the theoretically best path from being a chemically feasible synthesis plan after all. Also for hypergraphs, algorithms for finding the K shortest hyperpaths exist.

The focus of this project is to investigate algorithms for the K-shortest path problem on either standard graphs or hypergraphs.

Depending on the participants, this investigation may include:

- Theoretical study of algorithms, focusing on an exposition of their working, their correctness, and their asymptotic running times.
- Implementation of algorithms and experimental comparison of their running times on real data.

Topic ID: 14
Advisor: Peter Schneider-Kamp
Project title: Cloud infrastructure for drone inspections
Group Size: min 1 and max 4 students
Project description:
Drones offer exciting new ways of collecting image data on a wider variety of nature, culture, and infrastructure. Automated collection of such images requires advanced cloud infrastructure for planning, scheduling, routing, and path planning missions.
The aim of these projects is to review and evaluate appropriate state-of-the-art approaches for solving concrete drone inspection needs such as continuous deployment of cloud infrastructure, integration of satellite images, creation of 3D maps, and obstacle re-routing. This includes reviewing the relevant literature and tools as well as implementing a proof-of-concept for the given need. Different approaches should be empirically compared regarding their efficiency and effectivity.

Topic ID: 15
Advisor: Ruben Niederhagen
Project title: Crypto-Optimization for RISC-V
Group Size: min 1 and max 4 students
Project description:
RISC-V is an open instruction set architecture (ISA) based on RISC principles. Due to its open nature, there are several open source implementations of RISC-V (e.g. VexRiscv) that can be simulated on a PC, programmed onto an FPGA, or deployed to an ASIC and that can more or less easily be modified and extended.

Due to its compactness and the small resource footprint of RISC-V implementations, the RISC-V is particularly suitable for resource restricted embedded systems. However, for many applications in embedded security, a fast and efficient computation of cryptographic algorithms like encryption, signing, or hashing is demanded.

The first tasks of this project are to set up an existing implementation of the RISC-V ISA for simulation by installing the required tools like compiler, debugger, simulator, etc. on a Linux machine, to choose a cryptographic algorithm, to port it to the RISC-V, and to identify performance bottlenecks. The goal is then to implement optimizations for the cryptographic algorithm either in software using dedicated features of the standard RISC-V ISA or by implementing dedicated hardware accelerators or ISA extensions.

Topic ID: 16
Advisor: Jørgen Bang-Jensen
Project title: Data stream analysis using probabilistic techniques
Group Size: min 1 and max 4 students
Project description:
Data streams, such as searches on Google in a single day, can consist of enormous amounts of data - far more than can be stored and analyzed quickly on a single laptop.
Data Stream Algorithms are algorithms designed to extract information from streaming data. These algorithms do not have input data stored somewhere at the disk. Therefore, there is a limit to the number of times the algorithm can access some portion of data apart from having a limited memory space (generally sublinear in the number of elements of the stream) and hence they provide an approximate answer based on the summary of data they calculate.
These algorithms (usually) are aimed to calculate some properties over the given data stream. Examples of such properties could be approximate frequencies of the most common elements in the stream (e.g. which are the 100 most popular searches on Google today), the median of a large stream of numbers or the number of distinct elements in the stream. Often an approximate answer is good enough (e.g in data mining or learning contexts).
The purpose of the project is to study the use of various algorithms and data structures that are based on probabilistic techniques and randomization to problems related to data streams. Examples of such data structures are hashing, count (min) sketches and bloom filters.

The project can be purely theoretical, focusing on the theoretical aspects of the performance of the algorithms and data structures, or have a significant experimental part together with an account for the theory used.

The prerequisite is DM551 and an interest in probabilistic techniques such as hashing.

Topic ID: 17
Advisor: Marco Peressotti
Project title: Analysis of distributed systems and protocols
Group Size: min 1 and max 3 students
Project description:
Distributed systems play a crucial role in modern infrastructure, but are notoriously difficult to implement correctly, even for professionals. Although traditional tools for debugging and testing provide valuable insights in discovering bugs they cannot prove their absence since these tools cannot exhaustively cover the many possible concurrency issues, security weaknesses, and combinations of failures. Showing that distributed programs are correct requires more sophisticated techniques based on formal (semi)automated reasoning.

The goal of this project is to specify and analyse distributed programs or protocols using tools based on formal reasoning and adopted in industry [1] like TLA+ [2], ProVerif [3], AVISPA [4], and PRISM [5]. The subject of the analysis will be selected depending on the size of the group. The project requires familiarity with the basics of concurrent programming, operating systems, computer networks, and discrete methods.

[1] How Amazon Web Services Uses Formal Methods
https://cacm.acm.org/magazines/2015/4/184701-how-amazon-web-services-uses-formal-methods/fulltext
[2] https://lamport.azurewebsites.net/tla/tla.html
[3] https://prosecco.gforge.inria.fr/personal/bblanche/proverif/
[4] http://www.avispa-project.org/
[5] http://www.prismmodelchecker.org/

Topic ID: 18
Advisor: Konrad Krawczyk
Project title: Quantifying online news media coverage of health topics using text mining.
Group Size: min 1 and max 4 students
Project description: Online news media produce copious amounts of information that shape societal understanding/opinions on a variety of topics. A timely example is the communication of the development of the SARS-COV-2 pandemic and the measures to contain it that undoubtedly affect each individual's responses to the crisis. Using Text Mining techniques it is now possible to process vast amounts of online news data to create a quantitative picture of information received by vast swathes of the society. In this project you will employ Text Mining techniques to quantify the amount and type of health-related information in online news media.

You will work with ~10m news media items from landing pages from major outlets in the English-speakig world. You will employ a variety of text mining techniques, such as Named Entity Recognition, Sentiment Analysis etc. to quantify the news coverage on a range of health related topics. You will address the following issues:

- By developing a suitable topic model for ‘health news’ you will quantify what proportion of all news media items are health related.
- You will develop an unsupervised topic detection model to stratify the health news further by topics.
- You will quantify the sentiment of the news media on health-related topics.

The aim of the project is to get you familiarised with a variety of Text Mining algorithms, working on real-world data all-the-while addressing important questions in health sciences.

Topic ID: 19
Advisor: Jacopo Mauro, Fabrizio Montesi
Project title: Development of a Realistic Cloud-Native Web Application and its Evaluation
Group Size: min 3 and max 4 students
Project description:
The goal of this project is to develop a cloud-native implementation of the backend of the web application “Real World”, the reference specification for evaluating web development frameworks (https://github.com/gothinkster/realworld), using the service-oriented programming language Jolie (https://jolie-lang.org). The backend will be developed as a composition of microservices: small components that can be deployed and scaled independently on the cloud. The project includes implementing a deployment strategy for the web application on a major cloud framework, developing an automated testing framework, and evaluating the code of the web application against the numerous other implementations that already exist in other frameworks/languages (https://github.com/gothinkster/realworld#backends).

Topic ID: 20
Advisor: Rolf Fagerberg
Project title: Advanced Algorithms for the Minimum Spanning Tree Problem
Group Size: min 1 and max 4 students
Project description:
The minimum spanning tree (MST) problem is one of the oldest algorithmic problems known, with the first algorithm appearing in 1926, long predating any physical computer. The problem is simple to state and is taught already in the first year of the Computer Science programme (DM507), where several algorithms running in O(m log(m)) time are presented. Actually, a large number of more advanced algorithms have been given, with improved running times such as O(m loglog(m)) and O(m alpha(m)). To this day, the complexity of the problem is still open, although optimal algorithms with running time O(m) exist for the cases of randomized algorithms and of RAM algorithms for inputs with integer weights.

The goal of this project is to explore one of more of these more advanced algorithms, either theoretically, empirically, or both.

Prerequisite: a decent grade in DM507 and general interest in algorithms, their analysis, and/or their implementation.

Appendix

The Central Assignment Procedure

Input:

- N, the set of project topics
- M, the set of students
- Capacity limits of the advisors in terms of number of groups for each different size from 1 to 4
- N_i, for i in M, the ranked lists for each student of a subset of N, |N_i|>=5. The ranked list defines a strict total order on its elements.

Procedure:

1. A value v(i,j) in [1..|N_i|] is associated to every topic j in N_i for all students i in N such that v(i,j)=1 if j is the first (highest ranked) topic in the order of N_i, v(j)=2 if j is the second topic of N_i and so forth.
2. Let X be the set of all the assignments of topics to students that satisfy the capacity constraints and let x(i) be the topic that student i receives in assignment x in X. Further, let
   $$K = \min \{ \max \{v(i,x(i)) \mid i \text{ in } N\} \mid x \text{ in } X\}$$
   That is, let K be the best (smallest) value that the worst off student can receive in any assignment in X.
3. For k from K to 2:
   a. Determine the set of all feasible assignments X' subseteq of X that do not assign students to topics with value larger than k
   b. Remove from X' the assignments that do not minimize the number of students assigned to priority k
4. Among the assignments left select arbitrarily one.

If in the procedure above no feasible assignment can be found in step 2, that is, the set X is empty, then we will increase the capacity and repeat the step.
The first attempt will be with capacity limits of 2 groups of any size per advisor. Then, the following limits will be tried in the order halting at the first that admits a feasible solution:
- Max 2 groups of any size OR 3 individual students
- Max 2 groups of any size OR 4 individual students
- Max 2 groups of any size OR 5 individual students.