Constraint Modelling for Combinatorial Optimisation, 5.0 c

Course code: 1DL449, Report code: 11032, 33%, DAG, NML, week: 43 - 02 Semester: Autumn 2016

Result

This evaluation is answered by 53% (10/19) of the respondents.

Below are statistics on single- and multiple-choice answers and freeform text. Additionally, the summaries for freeform text responses that students will see are also shown.

1: Overall, how satisfied are you with the course?

Answers: 10

Answer options:

Median: 5 Mean: 4.7 / 5

0. Do not know/not relevant/do not wish to answer	0 responses	0%
1. Not at all	0 responses	0%
2.	0 responses	0%
3.	0 responses	0%
4.	3 responses	30%
5. Extremely	7 responses	70%

2: To what extent have you made the effort to benefit from the course content?

Answers: 10

Answer options:

Median: 4 Mean: 4.3 / 5

0. Do not know/not relevant/do not wish to answer	0 responses	0%
1. Not at all	0 responses	0%
2.	0 responses	0%
3.	1 responses	10%
4.	5 responses	50%
5. To a very high extent	4 responses	40%

3: This has been especially good about the course: Answers: 9

Projects in this course were very fun and talking about the concepts of things like P vs NP were fun and awesome.

Very interesting area. Presented a very useful way to view programming. Also a nice way to show the benefits of using different viewpoints when attempting to solve a problem.

[Trying different viewpoints in Assignment 2 was very good.] [Optimizing for a certain backend was good in Assignment 3 but needs more background material (difficult to know how to optimize for a certain backend with the current material).] [Deadline 18:00 is good.] [Project part was fun but exhausting. However, the course does not fill 5 hp without it.] [Solution sessions were really good.] [Gustav is a good fit as TA for this course, as he can give deeper insight into CBLS than most while also being able to answer most questions about other technologies.]

The lectures were outstanding as always with Pierre. The assignments were fun, with just the right amount of help and hints. Better facilities for extracting results generated by various MiniZinc solvers such as scripts would be good, which we have talked about.

I really liked that there was a solution session for assignment 3 after all. I liked the freedom we had on our projects

I didn't have much interaction with Pierre but I liked his lectures, very cooool... Gustav was very friendly and available. I liked the "unforgiving" nature of the course, high demands and expectations bring out the best in us lazy students. I have learned a few things not directly from the course but still relevant for it and other courses, such as linux/bash, scripting, github. Along with minizinc those are tools that I will have much use of.

The course and its lectures were very interesting and thought-through. The teachers were very friendly and helpful.

Exercises have excelled at being fair, challenging and provided in a timely manner. In no case did a task seem impossible provided the course material, however difficult it was. Pierre has been (as one would expect from other courses with his involvement) an excellent lecturer, both in terms of having knowledge and sharing it. Both in terms of a lecturer and assistant, all sessions with Gustav have been a delightful privilege. Overall, I would highly recommend this course due to its important content as well as its efforts to motivate students to gain understanding of it.

All parts. The project was nice!

4: This could be improved in the course: (Make your suggestions as constructive as possible)

Answers: 8

Since there's no textbook as a reference it would be great to be directed to some other online resources.

Minizinc feels fairly immature. Tooling, documentation and in part the language needs further work. The greatest pain point is probably "running several models over several problems with a timeout and getting sensible results when things blow up". Here I would suggest something on the lines of a platform independent script that can be trivially modified for the different assignments. The thing that took way too much time for me was in particular how to figure out a good way to stop a process and collect data from it.

[Be more clear what should be turned in for each assignment.] [Lecture pacing a bit slow (note that I have read COCP, had I not, I might've had a different opinion)]

A more detailed cheat sheet with more details about known difficulties with the various solvers would be welcome, so that when something doesn't work, it's more clear when to skip a given solver, etc.

Assignment 3 could have clearer instructions on task c (what is expected from the student)

Either the listed prerequisites must be "increased" or the course must include more introductory material. Coming from a math background with only minimal CS ("10 credits in programming") the course was very difficult not because of the mathematical concepts but because it seemed to be assumed that everyone knew CS, such as general Linux usage, writing scripts to run and record several experiments. These are not things you learn in "10 credits in programming". Although I enjoyed the lectures, they weren't very great for learning, unless you remember everything you hear or you're a proficient stenographer. Slides of course are not great for repeating stuff. I can understand why there was no book for the course, but some literature recommendations would have been nice.

The course would be improved by providing more information on different solver technologies, especially at an earlier stage. Finding useful information on how SMT solvers differ from SAT solvers, for example, was much more difficult than expected and lead us to rely on some of the earlier slides and lectures, that essentially only provide an overview. We struggled very much with understanding how CBLS should be properly used, and learned the most about it during solution sessions, at which point it naturally was too late for us to use the newly gained knowledge. Just by providing the topic slides for each solver by the time of the second assignment would have been very helpful. Another issue has been, much due to the nature of the course topic, a reliance on finding just the right combination of solutions for writing an efficient model -- this can however hardly be considered a critique as most emphasis seemed to have been placed on analysis rather than objective performance. Overall, the potential improvements that come to mind are hardly necessary for a course that has been, in all respects, excellent.

Too much time is spent on testing stuff / writing tests. Since the assignments are pretty standardised, a nice test framework which generates results based on predefined .mzn-filenames would be nice.

Summary of free-text responses/comments for the whole course evaluation