DM865

Course Organization

This course is about Discrete Optimization via:

- Approximation Algorithms
- Heuristics

Problems:

- Set Cover
- TSP
- SAT
- Knapsack
- Scheduling
- Bin Packing
- Bonus: the project problem

Techniques:

- LP-rounding
- Primal-Dual
- Randomized LP rounding
- Construction Heuristics
- Local Search
- Metaheuristics
- Implementation Framework
- Efficiency issues
- Experimental Analysis

Schedule

- Class schedule:
  - See course web page: dm865.github.io
  - mitsdu.sdu.dk
  - Changes can occur (reload web page often)

- Workload (approximately):
  - Each week when **not** working on the project:
    - 6 hours of intro phase + training phase
  - Each week when working on the project:
    - 2 hours of intro phase + training phase
    - 2 hours of questions / discussion of the project

Communication media

- Public Web Page [WWW] $\Leftrightarrow$ BlackBoard `e-learn.sdu.dk` [BB] (links from and to each others)
- Announcements in [BB]
- Course Documents in [BB] (unless linked from [WWW])
- Personal email `lenem@imada.sdu.dk`, `marco@imada.sdu.dk`
- Office visits
- Mid term evaluation in class

Literature

WS David P. Williamson and David B. Shmoys. The Design of Approximation Algorithms. Cambridge University Press. 2010.

MAK W. Michiels, E. Aarts and J. Korst. Theoretical Aspects of Local Search. Springer Berlin Heidelberg, 2007

Other articles

Evaluation (1/3)

- Two practical assignments:
  Part I:
  Local search and experimental analysis for a routing problem
  Part II:
  Builds up on Part I:
  Metaheuristics for a routing problem

- Oral exam in June.
  No preparation, External censor, 7-scale grade.

Evaluation (2/3)

- The final grade depends on the three parts.
- The assignments will be graded by the teacher
- The assignments will be available to the censor at the oral exam.
- Ten minutes of the oral exam on the assignments
  Ten minutes on the other part of the syllabus (approximation algorithms).
- The assignments should be carried out in pairs. We encourage the pairs to change between part 1 and 2.
- Communication between groups not allowed.

Evaluation (3/3)

- The assignments are meant for learning. There will be discussion classes where we will help you with your questions.
- You should be able to improve during the time of the course.
  Hence it matters most to us what you can do at the end.
- If you had trouble with one assignments you should still be able to get 12.
- If you have done well in the exercises but know nothing at the time of the exam you should still be able to fail.
- But this does not mean that the assignments should be taken lightly, they do have an influence and they are perhaps the part where you will really learn to solve problems.

Practical Assignments

- Really practical! $\rightsquigarrow$ programming in Python 3.
  (Refresh it when you have time).
- You will be provided with a starting framework but expect quite some work. The problem is quite challenging.
- It would be nice if you could use R for the analysis of data.
  (Some code examples might be provided but R will be useful in your career, learn it!)

Practical Assignments: Contents

- Algorithm design
- Modeling
- Implementation (deliverable and checkable source code)
- Written description
- (Analytical) and experimental analysis
- Performance counts!

Currently evaluating a web submission with automatic check, execution and comparison.

We hope you will have fun!

Active Participation

- We expect you to stay up-to-date with the course
  The literature is not mandatory to read, you can get along well with slides and lecture notes.
- Please ask questions!
- Experiment and explore
- Work with others
- Give us feedback.