How to get good Grades
– Would you like to know ???

Thomas Stidsen

thst@man.dtu.dk

DTU-Management
Technical University of Denmark

Outline

- How you should make your reports
- Testing ...
- Example report ...

DTU

The goals of this course

- Teach you as much as possible about metaheuristics
- Teach you how to report the findings in a scientific form.

How will I grade the course ?

Very simple: I will **MAINLY** grade you based on the report.

Hence this report is: Important, important important

I will also grade you based on the oral exam the 19/5 and based on your presentation of the project (either 2/5 or 9/5).

The Oral exam

In the oral exam I will exam you in the report you have handed in and in the different algorithms you apply.
The exam can also be used to correct errors in the report, i.e. if I discover an error in the report I will ask questions about it, and you get the chance to respond to the error ...

What is the purpose of the report ?

Forget for a moment that you (probably) want a good grade. The purpose of the report is:

- Transition of knowledge: The reader should become wiser.
- Reproducibility: Other persons should be able to reproduce your results based on the report.

These requirements correspond to the requirements of scientific articles and since this is a Ph.D. course, consider this to be your first exercise in making such an article.

How to make a report

What should a report contain ?

- Introduction
- Problem description
- Metaheuristic description
  - In general
  - Specialized
- Parameter tuning description
- Test description
- Test results
- Discussion and Conclusion

General Metaheuristic Description

What should you describe ?

- An introduction to the type of metaheuristic and a description of the principles of this particular type.
- The overall algorithm, preferably in pseudo code
- The meaning of the different parameters

Specialization Metaheuristic Description

What should you describe ?

- Any variations you invent/construct should be described **carefully**
- The adaptions made to specialize the metaheuristic for the particular problem should be descriped:
  - Neighborhood/Operators
  - Fitness calculation (often not so necessary)
  - Any other special adaptions.
- The meaning of the different parameters

Tests

Given two or more (variants of) algorithms how to find the best ?

- Parameter tune each algorithm
- Test algorithms
- Compare average performance

Parameter Tuning

Time is not a parameter: It is given externally!:

- Make a list over the parameters
- Make a list of test values for each parameter
- Select a small diverse data set for tuning
- Calculate average performance and spreading
- Select the best values ...

List of parameters

For Simulated Annealing:

- Start temperature
- Temperature decrease
- ....

In the report I want to see a **complete** list of the parameters for the algorithms (and neighborhoods). You may decide to fix some of the parameters, that is ok as long as you **describe** that in the report.

List of parameter values

For Simulated Annealing:

- Start temperature: $\{25, 50, 100\}$
- Temperature decrease: $\{0.85, 0.90, 0.95\}$

This selection of parameter values is critical.

Given a dataset

It is very important to have a good dataset to test on:

- A number of different data samples are needed here typically 6 to 10
- The dataset is divided into two parts:
  - A training dataset, usually rather small, 3-4 samples. These should be **representative**.
  - A test dataset, the rest.

Why ?

Why should the datasets be divided ?

- To avoid getting too good performance (for neural networks known as over learning).
- To reduce parameter tuning time ...

Averaging over different datasets

We have a problem: How can we compare performance on datasets with very different absolute values ? E.g. what is the average (minimization) performance (for a specific parameter setting) when we have two data sets and get the following performance:

1. $\{3, 15, 7\}$
2. $\{3002, 2950, 3030\}$

What we need to do is to calculate the **combined average** performance.

Averaging over different datasets II

Assume we know the optimal solution for each dataset: 2 for the first dataset and 2934 for the other dataset. Then we can calculate the percentage gap:

$$E = \frac{1}{|I|} \cdot \sum_i \frac{(z_i - z^*) \cdot 100}{z^*}$$

1. $\{3, 15, 7\}$: $E = 316.67$ and $\delta = 305.51$
2. $\{3002, 2950, 3030\}$: $E = 2.04$ and $\delta = 1.38$

These numbers we can then calculate averages for, i.e. $E_{tot} = 159.36$

When we have no optimal results ?

We have two (worse) options:

- Use a lower bound ...
- Use an approximation of the optimal result (the best solution found during the entire parameter tuning process ...), for that dataset

List of results of parameter tuning

	0.85	0.90	0.95
100	(Avg., δ)		
50			
25			

I want to see this matrix of the parameter tuning results in the report ...

Parameter Choice

How to select the best parameters ?

- Take the best avg. performance **AND**
- Take the lowest spreading

This is actually a multi-criteria optimization ! Use your common sense to make the best choice. When you have done this for all the meta-heuristic (variants), you are ready to test ...

List of results of parameter tuning

We assume minimization:

	0.85	0.90	0.95
100	(120.5, 2.4)	(110.4, 5.5)	(109.2, 1.9)
50	(133.1, 1.7)	(145.1, 1.3)	(113.7, 1.7)
25	(119.4, 3.4)	(130.2, 2.1)	(117.8, 2.2)

What is the best parameter value ? I would probably select $100, 0.95$ and hence get the $E = 109.2$ and $\delta = 1, 9$.

Test

Finally, given the best parameter settings, calculate avg. performance for each dataset in the test set, e.x.:

	Alg. 1.				Alg. 2			
Problem	Best	Mean	Variance	Time	Best	Mean	Variance	
scp51	257	259.60	2.15	5.5	263	263.00	0.00	
scp55	212	212.50	0.47	4.9	212	212.00	0.00	
Avg.								
Total Avg.								

Presenting the test

A few major points:

- **Always report average and spreading** for each algorithm for each test set.
- Best performance is not interesting
- Time is not interesting (it is fixed ...), but I want to know to what value you fixed it !
- Subtotals for different sizes of datasets should be presented. This may reveal that different performances depending on the data-set size and/or type.

Interpretation and discussion of tests

Important, but very problem/algorithm specific, what can you observe from the results ?

Pitfalls

A number of possible errors:

- **Time is not a parameter**
- Want to test too many parameters and too many values $\Rightarrow$ The tests takes (literally) 100 years ...
- Forget to split the dataset in two ...

Time

Very important to understand: For all heuristics there is a time-performance tradeoff: The longer the time the better the performance ! Hence:

- Time is an “externally” set parameter for your meta-heuristics
- For the projects you can decide the time yourself
- The time allocated per. run has to be the same for the parameter tuning and the testing
- Do not choose long test times ...

Many parameters and values

What to do ?

- Still list all parameters
- Fix certain parameters from the list which you assume to be non-important **but write that in the report**
- Start with few values for each parameter, then add more ...
- You **have** to have at least 2 parameters

If the best parameter in the test is an outer point, expand the interval ...

Overall goals of report I

The report I want to see should ideally be like a scientific article. Hence it is important that:

- The report is clear an consise in the language.
- The algorithms are precisely described, with relevant references.
- The algorithms are correctly parameter tuned.
- The algorithms are correctly tested.
- The results are commented and discussed.

Overall goals of report II

The overall idea is that the report should make its reader wiser and if the reader decides to repeat the experiments, all the details should be there and the tests should not be too far off, i.e. there should be little difference in the performance of the algorithms constructed and your results. This is what you should aim for and this is what I will grade people on. **The report should make it possible for one of your fellow students to repeat your experiments**

Negative results

What if your algorithm does not perform as well as expected ? Don't panic ! Negative results are very usefull ! Instead try to explain why the algorithm has problems. In fact, according to **Karl Popper** this is true science: Proving theories are wrong !!!

What not to do !

- “If I make the perfect program with the smartest programming tricks, I will get at good grade ...”
- “If I ask the lecturer he will think I am stupid ...”
- “The most interesting result is the best performance after 10 runs on one dataset, where I have fitted the parameters extremely well for that dataset ...”

What not to do ! II

- “The lecturer love to do statistics, hence I just give him all my numbers ...” **I dont do statistics**. All the calculations should be in the report.
- “The program is in the appendices, so if the lecturer is interested, he can just read it !” **I dont read appendices**, but they should be there !

Dont be scared !

The requirements described here may seem pretty ambitious, but it is only an ideal. Give it your best shot, ask for advice and things will turn out for the best. Furthermore this is a good exercise for a future master's thesis ...