Reporting Experimental Research

Guszti Eiben
Computational Intelligence Group, VU Amsterdam

Version: January 2021

In this document we briefly explain how to write good reports on experimental research. We assume an internal student research project in Computational Intelligence, such as an internal mini master project, bachelor project, master project, or a research assignment related to one of our courses. External master projects, i.e., internships at companies outside the VU, are not directly covered here. However, much of what we outline here is rather generic, applicable even for writing scientific papers for journals or conferences. Note that, while these guidelines are oriented at ***reporting*** research, they also give useful hints about ***conducting*** research.

1 Research target

A good research project has a target. This can be a moving target, changing as the project goes, but in the final report these historical changes need not be shown. A target can be formulated either as a Research Question (RQ), as a Research Goal (RG) or as a Hypothesis (maybe multiple hypotheses). These must be given in the beginning, in the section usually called **Introduction**. RQs and RGs are often convertible into each other, but once you make a choice you must stay consistent: an RQ needs to be answered, an RG needs to be assessed in the last section called **Conclusions**. That is, if you have RQs, the Conclusions must provide the answers, if you work with RGs you need to discuss if and to what extent you have achieved them. If the target is a Hypothesis, then the Conclusions must discuss whether it is true or not.

Example 1 Imagine that everybody solves problem X with a blue algorithm. Your target is to do this with a red algorithm. Then you can phrase the target as an RQ “Is it possible to solve problem X with a red algorithm?” or as an RG “To develop a red algorithm for solving problem X”. Your possible conclusions will differ accordingly: “The answer is yes, our experiments demonstrate that red algorithms can solve X in 90% of all test cases.” or “We have successfully developed a red algorithm that could solve X with a 90% success rate.’’ Alternatively, you may start with a Hypothesis “We hypothesize that a red algorithm outperforms the blue ones commonly used in the field”.

2 Report logic and experiment design

A well phrased RQ/RG gives a good hint about the possible (type of) conclusions. These can be formulated already in the beginning and used to help set up the experiments. The main logic is then twofold:

- to get such-and-such answers I need such-and-such data;
- to get such-and-such data I need such-and-such experiments and log files.

Thinking backwards from the possible (type of) conclusions can prevent spending much time on computer runs with an RQ/RG in mind and finding out at the end that the relevant data have not been logged.

Example 2 Imagine that you have the RQ “Is it possible to solve problem X with a red algorithm?” and your hypothesis (the thing you hope to evidence) is that “The answer is yes, our experiments demonstrate that red algorithms can solve X in 90% of all test cases.” Then you need to select some, say 100, instances of problem X as test cases, run your red algorithm on them, and log the success/failure of each run. Note that these log files will NOT be suited to conclude “The answer is yes, our experiments demonstrate that red algorithms can solve X faster than blue algorithms.” If that is your envisioned answer then you need to log the time-to-solution for each run instead of a simple success/failure. Furthermore, you need to run 200 experiments, 100 with your red algorithm and 100 with a blue algorithm.

These considerations can be instrumental in designing and setting up the right experiments. Therefore, they should be made before performing the actual computer runs.

3 Clarity and reproducibility

Whatever data you collect, present, and analyse two things must be made clear in the report. Firstly, how did you obtain these data? That is, what algorithm and what problem instances did you use, and what were the experimental conditions (e.g., the run time per test case, the exact parameter values used, the number of independent repetitions in case of a stochastic algorithm, etc.) Secondly, what do your numbers, tables, graphs, charts, whatnot exactly mean? To this end, you must clearly explain what ‘observables’ you monitored and logged. These form your raw data, e.g., for each single run you can monitor *q(t)* being the quality of the best solution known to the algorithm at time step t and decide to log it after every 10 time steps (t = 0, 10, 20, …) for further analysis. Furthermore, you must explain how these raw data were used to produce the aggregated figures. For instance, for each problem instance you may define an aggregated measure (not an observable, but calculated from the given observables!) called *success_for_red* as a Boolean value: 1 if the red algorithm had a better *q(t)* value at the end than the blue algorithm, 0 otherwise. These details should be presented before the results are shown. A common approach is to have a section called **Experimental Setup** preceding the section called **Experimental Results**. The description of the algorithm used can be part of the Experimental Setup section, but if the algorithm is newly invented then it is usually given its own section, e.g., **Algorithm Description**, before the Experimental Setup.

Example 3 Your Experimental Setup section should show the specific parameter values of the red algorithm. A good way to do this is using a table that shows the parameters of the algorithm and the specific values you gave them for your experiments. If you perform different experiments with different parameter values, you can add more columns to this table. Then you could identify 100 test problems by referring to an online repository and explaining that you used the first 50 instances of their collection of category A problems and the last 50 instances of their collection of category B problems. NB. You should argue why you did it like this. Furthermore, you can explain that you only log *q(t)* at termination, introduce the measure *success_for_red* as above, and define *success_rate_for_red* as the sum of *success_for_red* for all runs, divided by 100. You can do the same *_for_blue*. As the last step you can explain that 100 runs with red and 100 runs with blue were performed every day for a whole week. The Experimental Results section can then show 7 pie charts, one for each day of the week, exhibiting in *success_rate_for_red* as a red slice, *success_rate_for_blue* as a blue slice, and the ties in white. Alternatively, you may decide to split the results by problem category and show 14 pie charts, one for category A and one for category B for every day of the week.

With some forward thinking you can design the charts (tables, graphs, etc.) already before running the experiments. If you can write simple scripts that produce the charts from your raw data, then you just push the start button and wait till your charts are automatically filled. This prevents "tweaking" the outcomes afterwards and makes a smooth and efficient workflow. This is very important, because the first ideas (implementations, experimental scenarios, data plots, etc.) never work out as intended. Experiments bring new insights and these lead to reformulating the RQ or RG, and/or changing the algorithm, and/or changing the experimental setup, etc. This means that you have to iterate a lot, changing some details and re-run the experiments. Automating everything allows you to do more tests and gives you more time for thinking.

Last but not least, reporting all the details makes your research reproducible. It enables others to verify your findings independently, which is essential for good scientific conduct.

4 Related work and citations

Most academic research is embedded in the context of an existing field. An RQ/RG seldom falls out of the blue, hence there is always some relevant existing work. To position your work and to identify your contribution to the field (i.e., to tell the readers what is new here) this relevant work should be reviewed. Depending on how generous you are in defining "relevant", this could mean a structured discussion of 5 to 25 scientific articles (papers). It is good practice to devote a **Related Work** section to this, following the Introduction. An advisable rule of thumb here is "When in doubt, cite it!" Citations should not be given in footnotes on the same page, but in a separate section called **References** or **Bibliography**. This is typically the very last section of the whole report, only followed by the **Appendices** (if applicable). References come in various types, depending on the type of publication, and in all cases your bibliography item must be correct and complete. Completeness depends on the type, the most frequently used types are the following.

- Book: Author(s), Title, Publisher, Year.
  A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing, Springer, 2003.
- Journal paper: Author(s), Title, Name of Journal, Volume, Number, Pages, Year.
  A.E. Eiben and S.K. Smit, Parameter Tuning for Configuring and Analyzing Evolutionary Algorithms, Swarm and Evolutionary Computation, 1(1):19-31, 2011.
- Conference paper: Author(s), Title, Name of Conference, Publisher, Pages, Year. If the given conference proceedings have an editor, his/her name must be given as well.
  R.-J. Huijsman, E. Haasdijk, and A. E. Eiben, An On-line On-board Distributed Algorithm

for Evolutionary Robotics, in J.-K. Hao et al. (eds.), Proceedings of the 10th International Conference Evolution Artificielle, LNCS 7401, Springer, pp. 73-84, 2011.

5 Putting it all together

Putting this all together we obtain a quite universal report structure as follows:

1 Introduction

2 Related Work

3 Algorithm Description

4 Experimental Setup

5 Experimental Results

6 Analysis and Discussion

7 Conclusions

8 Bibliography

Of course, there are several variations to this structure. As mentioned above, the description of the algorithm might be just a subsection of **Experimental Setup**. Also, the presentation of the problem instances used for testing the algorithms could be upgraded from a subsection in **Experimental Setup** and given in a special section called **Test Problems**. In some cases The **Analysis and Discussion** of the outcomes is not presented in a separate section, but is integrated in the **Experimental Results**. In other cases, **Experimental Results** only exhibits the results in a well arranged graphical and/or tabular layout.

6 Miscellaneous

Figures: They must be readable also in a B/W print. Their caption can be brief (e.g., "Progress curves of q(t) for the red algorithm."), but then the main text should explain what they mean.

Tables: Mind the font size! Pdf's can be magnified on screen, paper prints not. Tables also have captions, just like Figures.

Appendix: Use it to present stuff that is too technical or too detailed for the main text. For instance, an Appendix is appropriate for big tables with raw data (aggregated charts in the main text), the Python code of your algorithm (pseudocode in the main text), or the results to compare different versions of your red algorithm (best version you really use shown in the main text).

Writing style: There are books about this. Let us mention only two things here. Choose between the "we-form" and the "I-form", as we did for this document. Do not use a historical narrative style explaining first I-did-this then I-did-that. Keep it factual and link text by logic and causal relation, not by chronology. For instance, "Initial tests showed that red does not work. Therefore, orange was used in the final series of experiments." is good. "First I spent 1000 CPU hours just to learn that red does not work. Then I tried orange and it turned out to work. So I use orange in the end and my final results are based on that." is not good.

Word vs. Latex: Use Latex. :-) Overleaf is your friend. (https://www.overleaf.com/)