Measurability and Reproducibility in Timetabling Research: State-of-the-Art and Discussion

Andrea Schaerf and Luca Di Gaspero

Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica
Università degli Studi di Udine
via delle Scienze 208, I-33100, Udine, Italy
email: `schaerf@uniud.it` `l.digaspero@uniud.it`

Abstract In this paper, we first illustrate the state-of-the-art in timetabling research w.r.t. two important research qualities, namely measurability and reproducibility, analyzing what we believe are the most important contributions in the literature. Secondly, we discuss some practices that, in our opinion, could contribute to the improvement on the two aforementioned qualities for future papers in timetabling research.
For the sake of brevity, we restrict our scope to university timetabling problems (exams, courses, or events), and thus we left out other equally-important timetabling problems, such as for example high-school, employee, and transportation timetabling.

1 Introduction

Thanks mainly to the PATAT conference series, researchers on timetabling problems have recently started to meet regularly to share experiences and results, more than in the past. This situation has the positive effect of generating both a common language and a common spirit that is the base ground for cross-fertilization of research groups in the timetabling community.

However, according to what we have seen in the recent PATAT conferences, the road for timetabling to become a well-established research community is still long. The main issue, in our opinion, is that most timetabling papers tend to describe the authors' specific problem and *ad hoc* solution algorithm without taking enough care of neither the *measurability* nor the *reproducibility* of the results. The reader is thus "left alone" to judge the quality of the paper, and to understand what can be learnt from it.

This issue is, to some extent, common to all the experimental areas of computer science and operations research, as clearly explained by Johnson in his seminal and fundamental paper [14]. Nevertheless, we believe that this is particularly true in timetabling research, probably because of its shorter tradition as a scientific community.

Regarding measurability (or comparability), we believe that several "research infrastructures" are necessary in order to create the ground for truly measurable

E. K. Burke, H. Rudová (Eds.): PATAT 2006, pp. 53–62. ISBN 80-210-3726-1.

results. Specifically, they range from common formulations, to benchmark instances, to instance generators, to solution validators, and others. Related to it, but somewhat complementary, is the issue of reproducibility. To this aim, aside the features just mentioned, it would be also necessary to create the conditions for sharing code and/or executables among researchers.

In this paper, we try to describe the state-of-the-art with respect to these crucial qualities of experimental research in timetabling, and we also present some personal opinions on how to proceed to improve on them. For the sake of brevity, we restrict our scope to university timetabling problems (exams, courses, or events), and we left out other equally-important timetabling problems, such as high-school, employee and transportation timetabling. Nevertheless, to some extent, the proposed guidelines can have a broader application to all timetabling domains.

In details, we first survey what, in our opinion, are the most important steps that have been pursued so far in timetabling research in terms of either measurability or reproducibility of results (Section 2) . Secondly, we propose our personal "best practices" for improving these two qualities in the timetabling research (Section 3). Our aim is to encourage both the authors to write research papers of high level in these important aspects and the reviewers to demand for it.

2 State of the art

In this section, we review the most remarkable contributions to the aim of creating the ground for the development of high quality measurable and reproducible research in timetabling. We first discuss the "standard" problem formulations, the benchmark instances (datasets), and the related file formats adopted. Next, we move to the comparison methods proposed, such as competitions and statistical tools. Finally, we discuss the issue of the objective validation of the proposed results.

2.1 Problem Formulations & Benchmark Instances

It is well known that timetabling problems vary not only from country to country, but also from university to university, and even in different departments of the same university the problem is not quite the same [23].

Nevertheless, throughout the years it has been possible to define common underlying formulations that could be used for the comparison of algorithms. In fact, a few basic formulations have become standards *de facto*, as they have been used by many researchers. Needless to say, standard formulations allow the researchers to compare their results and to cooperate for the solution. Furthermore, algorithms developed for more complex *ad hoc* formulations, can be tested on the basic standard ones so as to asses their objective quality.

For the EXAMINATION TIMETABLING problem (ETTP), Carter *et al* [7] provide a set of formulations which differ to each other based on some components of the objective function. They also provided a set of benchmark instances [6] extracted from real data. Formulations and benchmarks by Carter have stimulated a large body of research, so that many researches (see, e.g., [4,12,8]) have adopted one of the formulations of Carter (or a variant of them, creating a new standard as well), tested on the benchmarks, and also added new instances. For more complex formulations, additional data have been added by other researchers, in an arbitrary way. At present, all available instances and the corresponding best results (up to 2003) are published on the Web [17].

We call LECTURE TIMETABLING problem (LTTP), the problem of weekly scheduling a set of single lectures (or events). This problem differs from course timetabling (discussed below) because the latter is based on courses composed by multiple lectures, whereas lectures are independent. In fact, when a course is given in multiple lectures per week, some cost components are related to the way the lectures are placed in the week. On the contrary, this concept is totally absent in LTTP. The LTTP differs also from ETTP because it has completely different objectives (e.g., no isolated event vs. spreading exams).

The LTTP has been discussed in [22] and it has been the subject of the timetabling competition TTComp2002[1] [21]. The formulation proposed for TTComp2002 has also become quite standard, and many researchers have used it for their work (see, e.g., [16,9]). Twenty artificial instances were generated for the competition, and they are available from the TTComp2002 web page. In addition, a few other have been proposed (and made available via web) in [24].

As mentioned above, the COURSE TIMETABLING problem (CTTP), consists in the weekly scheduling of the lectures of a set of university course. Unfortunately, no standard formulation has emerged from the community for CTTP so far. Up to our knowledge, the only formulation available on the Web [11] together with a set of instances is the one proposed by ourselves in [13], along with 4 instances coming from the real cases (suitably simplified and made anonymous) in our university.

2.2 Data Format

For all the problems mentioned above, an important issue for the spreading in the community of a formulation is the data format. For all the formulations discussed above, the data format used is an *ad hoc* fixed-structure text-only one. For example, for TTComp2002 the input data comes in a single file containing the scalar values (events, rooms, room features, students), followed by the elements of the input arrays, one per line. The output format follows the same idea. For the ETTP the input format is also rather "primitive", with a fixed grammar and no formatting tags. Unfortunately, for this problem no output format has been specified in the original web page and paper.

[1] In the competition the problem is named CTTP, where C stands for course; but we believe this is quite misleading, because it deals with isolated lectures/events, rather than courses composed by many lectures. Therefore we prefer for this problem the name LTTP

The use of fixed-structure formats makes it easier to parse the input from any computer language, and for any (naive) programmer, but may be more difficult to be maintained and checked. For example, it happened that Carter's ETTP instances were replicated incorrectly on other web sites. This was due to the presence of a few newlines added in the files, that led to different (less constrained) instances. This unfortunate episode, that might have caused wrong results in some papers, would have been avoided if a structured format had been used.

On the other hand, a structured format, such as XML, would be more suitable in terms of flexibility, extensibility, and maintenance. A few structured format have been proposed in the literature, such as STTL [5,15] and TTML [19]. In [20], the authors go even beyond the language, proposing a multi-layer architecture for the specification and the management of timetabling problems. Up to our knowledge, however, these proposals have received a quite limited attention so far. This is probably due to the fact that researchers have normally little interest in the advantages of a structured language, and they prefer the quick-and-simple text-only version.

2.3 Comparison Methods & Competitions

The fair comparison of different algorithms and heuristics is well known to be a complex problem, and it has no simple and straightforward solution. In fact, in order to assess that an algorithm is "better" than another one it is necessary to specify not only the instances used, but also on which features they are compared (e.g., quality of the objective function, success rate, speed, ...). The question gets even more complicated in presence of randomized/stochastic algorithms, which add a degree on non-determinism in the solution process.

For the TTComp2002, the solution algorithms (provided as executables) were granted a maximum CPU time for their execution (based on a CPU benchmark, about 500 seconds on a recent computer) and they were evaluated only on the value of the objective function, averaged upon the 20 proposed instances. Unfeasible solutions where not considered, so that, in order to be admitted to the evaluation, participants had to find a feasible solution for all instances.

For stochastic algorithms, the participant had to ensure that their solver could produce the same solution when checked by the organization (by providing the seed of the random generator). In this situation, it is not clear how to apply the CPU time restriction and the choice of the organization was to grant the maximum time *for each single trial*. This was done to ensure reproducibility, although it had a drawback. The participants could take advantage of the so-called *Mongolian horde* approach: run as many trials as you can with no time limit and report only the best of all of them.

Up to our knowledge, the TTComp2002 has been the sole attempt in this respect. All other comparison are based on results published in the literature, which however often report only part of the necessary information (running times, number of trials, ...).

2.4 Result Validation

When some results are claimed in a research paper, the reader (or, more importantly, the reviewer) generally has to trust the author without any actual proof on the results. Although the possibility that the author is deliberately claiming a fake result is rare, cases in which the claimed results turned out to be wrong are relatively frequent. They are normally due to bugs in the code or misunderstandings in the formulation of the problem, typically the objective function.

For example, for the Graph Coloring problem, for the famous benchmark instance DSJC125.5 a 12-coloring has been claimed (and published) in 2002, whereas it has been successively proved that the minimum number of colors is 17.

Therefore the validation of the results claimed is clearly an important step toward the full reproducibility of the results. For the LTTP, in the TTComp2002, the validation of the results was done directly by the organizers, who asked all the participants to supply an executable that accepts a set of fixed command-line arguments.

For ETTP, unfortunately, no validation tool is available. Validation is currently based only on voluntary peer-to-peer interaction based on exchanges of solutions and values.

For our formulation of the CTTP, we have developed a web page [11] that allows the other researchers to download the problem formulation, the data format, and the benchmark instances. More importantly, everybody is allowed also to validate his/her own solutions, and to insert it among the results obtained for that instance. All results are automatically published on the web site along with the date and other information.

3 Proposals

In this section, we highlight some practices that, in our opinion, could contribute to the improvement on measurability and reproducibility for future papers in timetabling research. Part of what we propose here can be found also in [14], although we try to extract the advices by Johnson that we believe best suit to the current state of timetabling research.

3.1 Statistically Principled Comparison

One of the key issues of performance measurement (often underestimated) concerns the methods to deal with the random nature of many methods for obtaining a sound comparison of the different techniques. In the practice, this issue is often neglected and just some tendency indicators of the stochastic variables like mean values (and, more seldom, also standard deviations) in n runs (with $n \approx 10$) are provided. Furthermore, in a rather myopic view, these summary values are often advocated as the final word on the clear superiority of a technique over their competitors.

However, as it is common expertise of other research areas, when dealing with stochastic variables it is not correct to draw any conclusion only on the basis of single estimates but a principled statistical analysis on the behavior of the algorithm is needed (see, e.g., [1,27]). Even in the simplest cases of comparison of two means the analysis should include some kind of hypothesis testing (e.g., the t-test or the Mann-Whitney test for the parametric and the non-parametric case, respectively), that at least provides the reader with a (probability) measure of "confidence" in the result. For more complex settings further analyses could be carried on and the statistical tool-case is plenty of methods for correctly coping with several situations that arises in practice (see, e.g., [18]).

As an example, Birattari [2] has proposed a principled methodology for the comparison of stochastic optimization algorithms, called RACE, which comes out also as a software tool for the R statistical package [3]. This procedure, originally developed for the purpose of selecting the parameters of a single meta-heuristic, could be employed also in the case of the comparison of multiple algorithms by testing each of them on a set of trials. The algorithms that perform poorly are discarded and not tested anymore as soon as sufficient statistical evidence against them is collected.

This way, only the statistically proven "good" algorithms continue the race, and the overall number of tests needed to find the best one(s) is limited. Each trial is performed on the same randomly chosen problem instance for all the remaining configurations and a statistical test is used to assess which of them are discarded.

It is worth to notice that the statistical comparison of algorithms outlined in this section is based on the assumption of having access at previous results (or better at the code) of the different techniques involved in the comparison. This is clearly related to the issue of reproducibility of results that, in our opinion, can be achieved observing the guidelines described in the following.

3.2 Formulation, Data Format, Instances, and Results on the Web

As already mentioned, many papers in timetabling describe the modeling and the *ad hoc* solution of a new timetabling problem. For this kind of papers, in general we cannot expect that the authors make all the steps for obtaining full measurability and reproducibility such as, for example, publishing all the code. In fact, this would be quite a big work that would probably be too time-consuming for a researcher, aside possible employer's concerns. Nevertheless, we believe that there are a few actions that could contribute in these respects, which are not too expensive in terms of work.

First, the authors must state the problem clearly and exhaustively. If this is not possible in the paper for space reasons, the full formulation should be posted in an accompanying web site. Secondly, the authors should also post in the web site all the instances considered (changing names for privacy reasons, if necessary) in the study, along with all the necessary information accompanying them: data format, algorithms, results, and running times. Finally, the authors should post also the files containing their best solutions, so that other researcher can verify the actual results, and possibly use that solutions for further studies and improvements.

These actions would ensure comparability with the results on future research by other researchers or also by the same authors[2].

3.3 Web-based Problem Management System

Nowadays it is very common to see web sites that describe all aspects of either a specific problem, see e.g. [10,25], or a research area [26]. These web sites normally exhibit references to papers, people, problem formulations, benchmark instances, and supply other information.

Web sites are surely very useful for the community, and their presence is crucial for the quality of the research. Nevertheless, we believe that there is a further step to be made to this regard. Inspired by the well-known concept of CMS (*content management system*), we envision the idea of developing what we would call PMS (*problem management systems*). A PMS is a web application (rather than a web site) that should allow the users to interact with the application performing automatically all the following tasks:

Add results: New results are first validated, and then possibly inserted in the database along with time-stamp and other user-supplied information.

Add instances: Instances can be inserted at any moment. Researchers that are interested in the problem can be automatically informed by email of this kind of events.

Manage instance generation: Newly generated instances can be created automatically by users through interaction with an instance generator.

Analyze instances and results: Instances and results can be analyzed automatically so as to produce important indicators: constrainedness, similarity to other instances or other results, ...

Add general information: People, references, links, code, and other information can be added. Links would be validated periodically in an automatic way, and broken ones can be removed. References can also be imported from other sites.

Translate data: Input and output data can be translated in different formats so that coherent data can be proposed in different format to the community.

Organize on-line competitions: Competitions on specific instances and with registered participants and fixed deadlines can be organized automatically. Results can be reported immediately.

Visualize: Solutions can be visualized in graphical form to give an immediate picture of the features and the violations.

The interesting point is that information posted through the PMS would get on-line immediately in an automatic way. Obviously, a PMS needs to provide against possible malicious uses, and therefore some of the actions mentioned

[2] Many researchers (including ourselves!) experienced the frustration of loosing their solutions (or other data) for some of the problems they worked on.

above would need the approval of the administrator before becoming effective. This however would be just a Yes/No button, so that the administrator is pushed to answer shortly.

The PMS would also maintain historical data (through versioning systems), in such a way to be able to retrieve information eliminated by updates and deletions.

Acknowledgements

We wish to thank Marco Chiarandini for fruitful discussions about measurability and reproducibility of research results.

References

1. R. Barr, B. Golden, J. Kelly, M. Resende, and W. Stewart. Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, 1:9–32, 1995.
2. Mauro Birattari. *The Problem of Tuning Metaheuristics as Seen from a Machine Learning Perspective*. PhD thesis, Université Libre de Bruxelles, Belgium, 2004.
3. Mauro Birattari. *The RACE pacakge*, April 2005.
4. E. Burke and J. Newall. A multi-stage evolutionary algorithm for the timetable problem. *IEEE Transactions on Evolutionary Computation*, 3(1):63–74, 1999.
5. E. Burke, P. Pepper, and J. Kingston. A standard data format for timetabling instances. In E. Burke and M. Carter, editors, *Proc. of the 2nd Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-97), selected papers*, volume 1408 of *Lecture Notes in Computer Science*, pages 213–222, Berlin-Heidelberg, 1997. Springer-Verlag.
6. M. W. Carter. Carter's test data. URL: `ftp://ftp.mie.utoronto.ca/pub/carter/testprob/`, 2005. Viewed: June 1, 2006, Updated: June 7, 2005.
7. M. W. Carter, G. Laporte, and S. Y. Lee. Examination timetabling: Algorithmic strategies and applications. *Journal of the Operational Research Society*, 74:373–383, 1996.
8. S. Casey and J. Thompson. Grasping the examination scheduling problem. In Edmund Burke and Patrick De Causmaecker, editors, *Proc. of the 4th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2002), selected papers*, volume 2740 of *Lecture Notes in Computer Science*, pages 232–244, Berlin-Heidelberg, 2003. Springer-Verlag.
9. M. Chiarandini, M. Birattari, K. Socha, and O. Rossi-Doria. An effective hybrid approach for the university course timetabling problem, 2004. Accepted for publication on the Journal of Scheduling.
10. Joseph Culberson. Graph coloring page. URL: `http://www.cs.ualberta.ca/~joe/Coloring/`, 2006. Viewed: June 1, 2006, Updated: March 31, 2004.
11. Luca Di Gaspero, Alessandro Fontanel, and Andrea Schaerf. Educational timetabling uniud. URL: `http://www.diegm.uniud.it/satt/projects/EduTT/`, 2006. Viewed: June 1, 2006, Updated: May 30, 2006.
12. Luca Di Gaspero and Andrea Schaerf. Tabu search techniques for examination timetabling. In E. Burke and W. Erben, editors, *Proc. of the 3rd Int. Conf. on*

the Practice and Theory of Automated Timetabling (PATAT-2000), selected papers, volume 2079 of *Lecture Notes in Computer Science*, pages 104–117. Springer-Verlag, Berlin-Heidelberg, 2001.
13. Luca Di Gaspero and Andrea Schaerf. Multi-neighbourhood local search with application to course timetabling. In Edmund Burke and Patrick De Causmaecker, editors, *Proc. of the 4th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2002), selected papers*, volume 2740 of *Lecture Notes in Computer Science*, pages 262–275, Berlin-Heidelberg, 2003. Springer-Verlag.
14. D. S. Johnson. A theoretician's guide to the experimental analysis of algorithms. In M. H. Goldwasser, D. S. Johnson, and C. C. McGeoch, editors, *Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges*, pages 215–250. American Mathematical Society, 2002. Available from `http://www.research.att.com/~dsj/papers.html`.
15. Jeffrey H. Kingston. Modelling timetabling problems with STTL. In E. Burke and W. Erben, editors, *Proc. of the 3rd Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2000), selected papers*, volume 2079 of *Lecture Notes in Computer Science*, pages 309–321. Springer-Verlag, Berlin-Heidelberg, 2001.
16. Philipp Kostuch. The university course timetabling problem with a three-phase approach. In Edmund Burke and Michael Trick, editors, *Proc. of the 5th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2004), selected papers*, volume 3616 of *Lecture Notes in Computer Science*, pages 109–125, Berlin-Heidelberg, 2005. Springer-Verlag.
17. Liam Merlot. Public exam timetabling data sets. URL: `http://www.or.ms.unimelb.edu.au/timetabling`, 2005. Viewed: June 1, 2006, Updated: October 13, 2003.
18. D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, sixth edition, 2005.
19. E. Özcan. Towards an XML-based standard for timetabling problems: TTML. In G. Kendall, E. Burke, S. Petrovic, and M. Gendreau, editors, *Proc. of the 1st Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA-03), selected papers*, pages 163–185. Springer, 2005.
20. Y. Lu P. D. Causmaecker, P. Demeester and G. Vanden. Using web standards for timetabling. In *Proc. of the 4th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2002)*, pages 238–257, 2003.
21. Ben Paechter, Luca Maria Gambardella, and Olivia Rossi-Doria. International timetabling competition, web page. URL: `http://www.idsia.ch/Files/ttcomp2002/`, 2003. Viewed: June 1, 2006, Updated: July 10, 2003.
22. Olivia Rossi-Doria et al. A comparison of the performance of different metaherustic on the timetabling problem. In Edmund Burke and Patrick De Causmaecker, editors, *Proc. of the 4th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT-2002), selected papers*, volume 2740 of *Lecture Notes in Computer Science*, pages 329–351, Berlin-Heidelberg, 2003. Springer-Verlag.
23. Andrea Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127, 1999.
24. Krzysztof Socha, Joshua Knowles, and Michael Sampels. A MAX-MIN ant system for the university timetabling problem. In *Proc. of the 3rd International Workshop on Ant Algorithms (ANTS 2002)*, volume 2463 of *Lecture Notes in Computer Science*, pages 1–13, Brussels, Belgium, 2002. Springer-Verlag, Berlin, Germany. Data available from: `http://iridia.ulb.ac.be/~msampels/tt.data/`.

25. Michael Trick. Challenge traveling tournament instances, web page. URL: http://mat.gsia.cmu.edu/TOURN/, 2005. Viewed: June 1, 2006, Updated: May 18, 2006.
26. The web site of the EURO working group on automated timetabling (watt). URL: http://www.asap.cs.nott.ac.uk/watt/, 2002. Viewed: June 1, 2006, Updated: November 28, 2005.
27. Jiefeng Xu, Steve Chiu, and Fred Glover. Fine-tuning a tabu search algorithm with statistical tests. *Int. Transactions on Operational Research*, 5(3):233–244, 1998.