OPTION: OPTImization Algorithm Benchmarking ONtology

Ana Kostovska
Jožef Stefan Institute &
Jožef Stefan International
Postgraduate School
Ljubljana, Slovenia

Diederick Vermetten
Leiden Institute for Advanced
Computer Science
Leiden, The Netherlands

Carola Doerr
Sorbonne Université, CNRS, LIP
Paris, France

Sašo Džeroski
Jožef Stefan Institute &
Jožef Stefan International
Postgraduate School
Ljubljana, Slovenia

Panče Panov
Jožef Stefan Institute &
Jožef Stefan International
Postgraduate School
Ljubljana, Slovenia

Tome Eftimov
Jožef Stefan Institute
Ljubljana, Slovenia

ABSTRACT

Many platforms for benchmarking optimization algorithms offer users the possibility of sharing their experimental data with the purpose of promoting reproducible and reusable research. However, different platforms use different data models and formats, which drastically inhibits identification of relevant data sets, their interpretation, and their interoperability. Consequently, a semantically rich, ontology-based, machine-readable data model is highly desired.

We report in this paper on the development of such an ontology, which we name OPTION (**OPTI**mization algorithm benchmarking **ON**tology). Our ontology provides the vocabulary needed for semantic annotation of the core entities involved in the benchmarking process, such as algorithms, problems, and evaluation measures. It also provides means for automated data integration, improved interoperability, powerful querying capabilities and reasoning, thereby enriching the value of the benchmark data. We demonstrate the utility of OPTION by annotating and querying a corpus of benchmark performance data from the BBOB workshop data – a use case which can be easily extended to cover other benchmarking data collections.

CCS CONCEPTS

• **Computing methodologies** → **Ontology engineering**; **Reasoning about belief and knowledge**; **Continuous space search**;

1 INTRODUCTION

Evolutionary computation, and black-box optimization in general, is a rapidly growing field, which has seen tremendous progress in the last few decades, where many new algorithms are developed daily, making it impossible for researchers and practitioners in the field to stay up to date with all these developments and difficult to share their results with the research community.

There are unfortunately no common standards for *which* performance data to record, nor *how* to store it. Storing, sharing, and making the benchmark optimization data reusable is challenging, because of the existence of different data formats that are weakly compatible. There exist a large number of different benchmarking platforms for optimization, which have mostly been designed independently from each other.

One well-established solution to the problem of managing such complex data pools is the use of ontologies. Ontologies are specifications of a shared conceptualization of data from distributed and heterogeneous systems and databases, and as such, they enable data interoperability, efficient data management and integration, and cross-database search.

2 THE OPTION ONTOLOGY

In this paper, we present the **OPTimization Algorithm Benchmarking ONtology (OPTION)**. The ontology we propose was designed with the main goal of standardizing and formalizing knowledge from the domain of benchmarking optimization algorithms, where an emphasis was put on the representation of data from the benchmark performance space. OPTION offers a comprehensive description of the domain, covering the benchmarking process, as well as the core entities involved in the process, such as optimization algorithms, benchmark problems, evaluation measures, etc. The OPTION ontology is publicly available on BioPortal[1] the largest repository of ontologies. We have also constructed a semantic annotation pipeline for annotating performance data, as well as a querying endpoint for querying the knowledge base. Finally, we have annotated benchmark performance data from 32 optimization algorithms and show how these annotations can be queried.

3 USE CASE

To demonstrate the utility of the OPTION ontology for semantic annotation, we consider the BBOB benchmark suite as a use case. We use a subset of BBOB data that includes the algorithms from the 2015-2017 competitions (three years). This gives us a total of 32 algorithms, which we semantically annotate using the OPTION ontology. All annotations were produced with the Apache Jena RDF library in the form of RDF graphs. For each of the 32 algorithms, a

---

[1]OPTION at BioPortal: https://bioportal.bioontology.org/ontologies/OPTION-ONTOLOGY
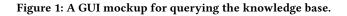
separate RDF graph was generated and uploaded to Apache Jena TDB2, a native triple store that efficiently stores RDF data.

The data in the ontology provides the full information on the used problem/dimension/instance/algorithm, as well as the corresponding performance data. In addition, data provenance information has been manually collected and linked to the performance data to trace its origin. The stored data provenance information includes the digital object identifier (DOI) of the paper, the paper's title, the authors' name, and the year of the publication.

4 QUERYING THE KNOWLEDGE BASE

After populating the knowledge base, we can use the SPARQL query language to ask queries. To enable this functionality, we have set up an Apache Jena Fuseki2 server and implemented two services. The query service provides an endpoint for handling SPARQL queries in a RESTful manner, while the upload service enables the upload of RDF data into the Apache Jena TDB2 triple store. A non-exhaustive list of queries that are currently supported by the OPTION knowledge base is as follows: (1) For a given algorithm (e.g., CMA-ES) and fixed budget scenario (e.g., between 1000 - 2000 function evaluations), return the noise-free fitness - Fopt calculated for a set of benchmark problems (e.g., f1 and f7) and a set of problem instances (e.g., the first 5 instances); (2) For a given algorithm (e.g., CMA-ES) and fixed target, return the number of function evaluations needed to reach the target; (3) For multiple algorithms and fixed budget return the noise-free fitness - Fopt calculated for a set of benchmark problems (e.g., f1-f5); (4) For a set of instances from a given benchmark problem, return all noise-free fitness - Fopt values for all (or a set of) algorithms with a fixed budget of 1000 function evaluations; (5) For a given study, return the provenance information.

Figure 1: A GUI mockup for querying the knowledge base.

SPARQL queries can become very complex and sometimes are seen as a bottleneck to the wider acceptance of Semantic Web technologies. SPARQL query construction is an error-prone and time-consuming task, which requires expert knowledge of the whole stack of semantic technologies. To facilitate the process of querying and to avoid the need of directly writing SPARQL queries, we propose the use of a GUI (for which a mock-up is depicted in Figure 1) to alleviate the problem explained above. Such a GUI would provide end-users with a search tool through which they can parameterize the predefined queries, without knowing the exact semantic data model used for the annotations. The users would construct the query by providing the search parameters of interest via input fields or predefined drop-down menus that contain ontology terms and the SPARQL queries would be generated in the background.

5 CONCLUSIONS & FUTURE WORK

The development of the OPTION ontology is a big step forward in the process of making benchmark data more reusable and interoperable. By annotating a large subset of the BBOB-data, we have shown the potential of the ontology to aid with data integration, while simultaneously providing powerful querying capabilities for direct analysis of the required datasets. This significantly reduces the time needed to collect data across many functions and algorithms, while being flexible in the type of performance perspective (i.e., fixed-budget, fixed-target) to be used.

Our next steps are as follows. First, we will annotate more data sets, starting with those provided by IOHprofiler [1, 3] and those of Nevergrad [2]. Further extensions include a web-based GUI, similar to what is shown in Figure 1. We believe this to be important for better adoption of our ontology by the optimization heuristics community, whose members are not necessarily familiar with SPARQL queries. This can then be further extended to include direct integration with existing post-processing tools, e.g., IOHanalyzer [4], which would greatly simplify the process of robust algorithm comparison.

REFERENCES

[1] Carola Doerr, Furong Ye, Naama Horesh, Hao Wang, Ofer M. Shir, and Thomas Bäck. 2020. Benchmarking discrete optimization heuristics with IOHprofiler. *Applied Soft Computing* 88 (2020), 106027. https://doi.org/10.1016/j.asoc.2019.106027
[2] Jérémy Rapin and Olivier Teytaud. 2018. Nevergrad - A gradient-free optimization platform. https://GitHub.com/FacebookResearch/Nevergrad. (2018).
[3] Diederick Vermetten, Hao Wang, Carola Doerr, and Thomas Bäck. 2020. Integrated vs. sequential approaches for selecting and tuning CMA-ES variants. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO'20)*. ACM, 903–912. https://doi.org/10.1145/3377930.3389831
[4] Hao Wang, Diederick Vermetten, Furong Ye, Carola Doerr, and Thomas Bäck. 2020. IOHanalyzer: Performance Analysis for Iterative Optimization Heuristic. *CoRR* abs/2007.03953 (2020). arXiv:2007.03953 https://arxiv.org/abs/2007.03953