OWL Ontology Optimization using Constrained Multi-objective Genetic Algorithm for Reducing Load of Inference Enabled SPARQL Query Execution

Naoki Yamada[1] and Naoki Fukuta[2]

[1] Department of Informatics, Graduate School of Integrated Science and Technology, Shizuoka University, Japan
yamada.naoki.17@shizuoka.ac.jp
[2] Department of Informatics, Shizuoka University, Japan
fukuta@inf.shizuoka.ac.jp

Abstract. On an inference-enabled Linked Open Data(LOD) endpoint, usually a query execution takes longer than on an LOD endpoint without inference engine due to its processing of reasoning. In optimizing an OWL ontology for executing SPARQL queries on an LOD endpoint with inference engine, there are a number of SPARQL queries that we would probably take into account and many of these factors possibly work against each other. In this paper, for reducing query execution time on an inference-enabled LOD endpoint, we propose an idea and its implementation to employ a constrained multi-objective evolutionary approach to make modification of ontologies based on the past-processed queries and their results. We employ a multi-objective evolutionary approach to realize better promoting and preserving diversity within the population. We also employ constraint handling to dealing with invalid solution such as inconsistent ontology. We show how the approach works well on implementing an inference-enabled LOD endpoint by a cluster of SPARQL endpoints.

Keywords: Semantic Web · SPARQL · Inference.

1 Introduction

Linked Open Data (LOD) are retrieved by a query written in the standard query language called SPARQL[3] for the retrieval of RDF data stored in an endpoint. Reasoning on LODs allows such queries to obtain implicitly stated knowledge or data from the given explicit relations. These techniques are important to make intelligent agents accessible to data on its external world. Techniques to utilize reasoning capability based on ontology have been developed to overcome several issues, such as higher complexity in worst case[1, 4–7]. We have presented an approach to cut off the queries which put unusual heavy loads to the reasoning

[3] http://www.w3.org/TR/rdf-sparql-query/

engine [11] as well as modifying the queries and ontologies to approximate such a heavy load queries [9, 10]. However, due to the complexity of queries and strong dependency on the features of the queries, it is not easy to implement 'one-fits-all' solutions for all possible queries on the endpoint. In this paper, to overcome this issue, we propose an idea and its implementation to employ a constrained multi-objective evolutionary approach to make modification of ontologies based on the past-processed queries to realize better promoting and preserving diversity within the population.

2 GA-Based Ontology Modification

We employ GA-based modification approach to approximate an ontology for an inference-enabled LOD endpoint [9]. In this paper, we use the query execution result to calculate the fitness value. It is time-consuming to relaunch an endpoint every query execution for shelter a result of query execution from the effect of other query execution cache. On the implementation of our GA, the number of times of relaunching an endpoint and the number of times of query execution will both be increased in proportion to a population size and the number of generations. Therefore, finding the optimal solution of an ontology modification method in a small population size and a small number of generations is important for reducing the time cost of GA.

Expressing an OWL ontology itself as a gene is a simple way to applying genetic algorithm to an OWL ontology modification. For example, assuming that we express ontology modification with 0 and one, if the value corresponding to an axiom is 0, delete its axiom from the OWL ontology, if the value corresponding to an axiom is 1, do nothing to the OWL ontology. It is difficult to apply the solution obtained by this method to other ontologies.

In our genetic algorithm, we express a modification rule for an OWL ontology as a gene. To find better solutions with an earlier generation than the result of applying a normal genetic algorithm, we use constraint handling and multi-objective optimization with genetic algorithm.

In the multi-objective algorithm that work under the principle of domination, fitness values of the n-objective functions are expressed as n-dimensional vector $\overline{a}$. The domination between two individuals $A \succeq B$ can be defined as equation (1) [3]. In multi-objective evolutionary algorithms that work under the domination principle, it aims to calculate the domination and obtain a set of non-dominated solutions [3].

$$\begin{aligned} A \succeq B \Leftrightarrow & a_i \geq b_i (\forall i \in \{1, \cdots, n\}) \text{ and} \\ & a_i > b_i (\exists i \in \{1, \cdots, n\}) \end{aligned} \tag{1}$$

3 Discussion

We set up a SPARQL endpoint using Joseki (v3.4.4) in conjunction with server-side Pellet [8] reasoner to enable OWL-level inference capability on the endpoint.

We used the dataset used in the conference track on Ontology Alignment Evaluation Initiative 2013 (OAEI 2013)[4]. Here we used Linklings ontology from the OAEI dataset in the preliminary experiment.

Fitness values for individuals are set to equation (2). Here, V_o is a fitness value of modified ontology o. F_o is a F-measure of the precision and recall on their output results over the modified ontology o. T_O is a query execution time of the original ontology O. T_o is a query execution time of the modified ontology o. In this experiment, we use $\alpha = 0.7$ and $\beta = 0.3$, respectively.

$$V_o = \alpha F_o + \beta \frac{T_O - To}{T_O} \text{ such that } \alpha + \beta = 1 \tag{2}$$

Figure 1 shows that maximum values of each objective over generations. Each objective value was plotted based on the evalution value (e.g, fitness value) of some specific queries on their execution results. In this case, objective3 and objective4 do not converge well in these number of generations.

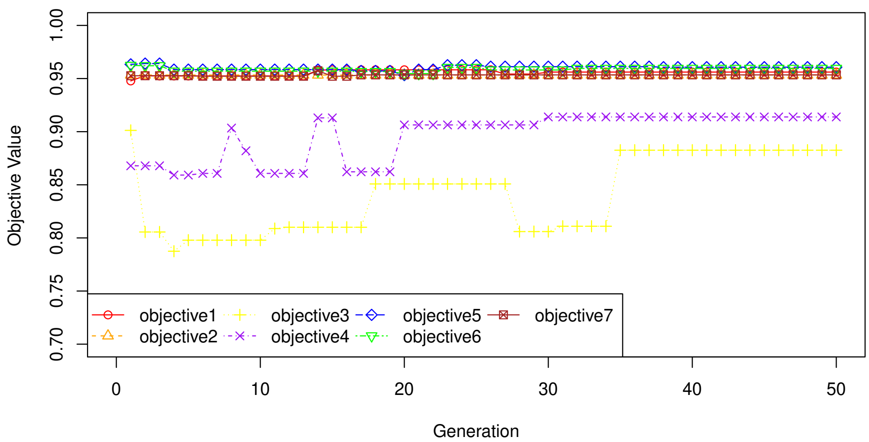

Fig. 1. Maximum values of each objective over generations

4 Conclusion

In this paper, we proposed an approach to employ multi-objective evolutionary algorithms that works under the principle of domination. For further improve-

[4] http://oaei.ontologymatching.org/2013/conference/index.html

ment of approximation performance, applying techniques often used in many-objective evolutionary algorithms such as the NSGA-III [2] is one of our future work.

Acknowledgment

The work was partly supported by the JST CREST JPMJCR15E1.

References

1. Baumgartner, P., Furbach, U., Niemelä, I.: Hyper Tableaux. In: Alferes, J.J., Pereira, L.M., Orlowska, E. (eds.) Logics in Artificial Intelligence. LNCS, vol. 1126, pp. 1–17. Springer-Verlag (1996)
2. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (Aug 2014). https://doi.org/10.1109/TEVC.2013.2281535
3. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing, pp. 195–213. Springer Publishing Company, Incorporated, 2nd edn. (2015)
4. Gonçalves, R.S., Parsia, B., Sattler, U.: Performance Heterogeneity and Approximate Reasoning in Description Logic Ontologies. In: Cudré-Mauroux, P., et al. (eds.) The Semantic Web–ISWC 2012 Part I. LNCS, vol. 7649, pp. 82–98. Springer-Verlag (2012)
5. Kang, Y.B., Li, Y.F., Krishnaswamy, S.: Predicting Reasoning Performance Using Ontology Metrics. In: Cudré-Mauroux, P., et al. (eds.) The Semantic Web–ISWC 2012 Part I. LNCS, vol. 7649, pp. 198–214. Springer-Verlag (2012)
6. Motik, B., Shearer, R., Horrocks, I.: Hypertableau Reasoning for Description Logics. Journal of Artificial Intelligence Research **36**, 165–228 (2009)
7. Romero, A.A., Grau, B.C., Horrocks, I.: MORe: Modular Combination of OWL Reasoners for Ontology Classification. In: Cudré-Mauroux, P., et al. (eds.) The Semantic Web–ISWC 2012 Part I. LNCS, vol. 7649, pp. 1–16. Springer (2012)
8. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical owl-dl reasoner. Web Semantics: Science, Services and Agents on the World Wide Web **5**(2), 51 – 53 (2007). https://doi.org/https://doi.org/10.1016/j.websem.2007.03.004, http://www.sciencedirect.com/science/article/pii/S1570826807000169, software Engineering and the Semantic Web
9. Yamada, N., Yamagata, Y., Fukuta, N.: Query rewriting or ontology modification? considering reasoning capability on lod endpoints. In: IEEE International Conference on Agents (IEEE ICA 2016). pp. 19–24 (2016)
10. Yamada, N., Yamagata, Y., Fukuta, N.: Query rewriting or ontology modification? toward a faster approximate reasoning on lod endpoints. IEICE Transactions on Information and Systems **E100.D**(12), 2923–2930 (2017). https://doi.org/10.1587/transinf.2016AGP0010
11. Yamagata, Y., Fukuta, N.: An Approach to Dynamic Query Classification and Approximation on an Inference-enabled SPARQL Endpoint. Journal of Information Processing **23**(6), 759–766 (2015)