Genetic Algorithms for RDF Chain Query Optimization

Alexander Hogenboom Viorel Milea Flavius Frasincar Uzay Kaymak

Erasmus School of Economics, Erasmus University Rotterdam
P.O. Box 1738, 3000 DR Rotterdam, the Netherlands
{hogenboom, milea, frasincar, kaymak}@ese.eur.nl

The full version of this paper, entitled RCQ-GA: RDF CHAIN QUERY OPTIMIZATION USING GENETIC ALGORITHMS *appeared in: Proceedings of the Tenth International Conference on E-Commerce and Web Technologies (EC-Web 2009), pages 181–192, Linz, Austria, 2009.*

Abstract

The application of Semantic Web technologies in an Electronic Commerce environment implies a need for good support tools. Fast query engines are required for efficient real-time querying of large amounts of data, usually represented using RDF. We focus on optimizing a special class of SPARQL queries: RDF chain queries. We devise a genetic algorithm, RCQ-GA, that determines the order in which joins need to be performed for an efficient evaluation of RDF chain queries. The approach is benchmarked against a two-phase optimization algorithm, previously proposed in literature. The more complex a query is, the more RCQ-GA outperforms the benchmark in solution quality, execution time needed, and consistency of solution quality. When the algorithms are constrained by a time limit, the overall performance of RCQ-GA compared to the benchmark improves even further.

1 Introduction

Semantic Web technologies are promising enablers for large-scale knowledge-based systems in an Electronic Commerce environment as they facilitate machine-interpretability of data through effective data representation. Fast query engines are needed in order to efficiently query large amounts of data in real-time environments, usually represented using the Resource Description Framework (RDF). RDF sources can be queried using SPARQL. The execution time of a query depends on the order in which parts of the query paths are executed. In the context of the Semantic Web, two-phase optimization (2PO) has been proposed to optimize RDF query paths [5]. However, other algorithms have not yet been used for RDF query path determination, while genetic algorithms (GAs) have proven to be more effective than SA in cases with similar characteristics [3]. Furthermore, GAs have proven to generate good results in traditional query execution environments [4]. The main goal we pursue consists of investigating whether an approach based on GAs outperforms 2PO in RDF query path determination. As a first step, we focus on the performance of such algorithms when optimizing a special class of SPARQL queries, RDF chain queries, on a single source.

2 A Genetic RDF Query Path Determination Algorithm

An RDF model is a collection of RDF facts declared as a collection of triples, each of which consists of a subject, a predicate, and an object. These triples can be visualized using an RDF graph, which is a node and directed-arc diagram, in which each triple is represented as a node-arc-node link. The RDF queries we consider are a specific subset of SPARQL queries, i.e., chain queries, where the WHERE statement only contains a set of chained RDF node-arc-node patterns. The order in which these node-arc-node patterns are joined affects the time needed for executing the query. The challenge is to determine the right join order, hereby optimizing the overall response time (i.e., minimizing the solution costs).

We propose to optimize RDF Chain Queries using a Genetic Algorithm: RCQ-GA. A GA is an optimization algorithm simulating biological evolution according to the principle of survival of the fittest. A set of chromosomes, representing solutions, is exposed to evolution, consisting of selection (choosing individual chromosomes to be part of the next generation), crossovers (creating offspring by combining selected chromosomes), and mutations (randomly altering selected chromosomes). Evolution is simulated until the maximum number of iterations is reached or several generations have not yielded any improvement. A chromosome's fitness can depend on its costs or its rank. In order for a GA to be applicable in RDF query path determination, the challenge is to find a balance between execution time and solution quality. We adopt the settings best performing in [4], but in an attempt to enforce quicker convergence of the model, we adapt their algorithm, BushyGenetic (BG), by allowing less generations without improvement and by reducing the population size. We also replace ranking-based selection with fitness-based selection.

We test our algorithm on a single source, generated using QMap [2], by assessing the performance of 2PO as proposed in [5], the BG algorithm [4], and RCQ-GA. We also assess the impact of a time constraint on 2PO and RCQ-GA; we employ a time limit of 1 second, as this allows the algorithms to perform at least a couple of iterations and we assume this to be an acceptable maximum waiting time in a real-time environment. Each algorithm is tested on chain queries varying in length from 2 to 20 predicates. Each experiment is iterated 100 times. Overall, the BG algorithm needs the most execution time. For relatively small chain queries containing up to about 10 predicates, 2PO turns out to be the fastest performing optimization algorithm. For bigger chain queries, RCQ-GA is the fastest performing algorithm. Furthermore, BG and RCQ-GA tend to find better solutions of more consistent quality than 2PO does with respect to the average costs associated with optimized chain query paths, especially for larger queries. When a time limit is set, a GA tends to generate solutions of even better quality compared to 2PO. The consistency in solution quality of RCQ-GA, as opposed to 2PO, is not clearly affected by a time limit.

3 Conclusions

In optimizing query paths for chain queries in a single-source RDF query execution environment, the performance of a GA compared to 2PO is positively correlated with solution space complexity and environmental restrictiveness (a time limit). In this paper it is shown that in the context of RDF chain queries optimization a GA outperforms 2PO in solution quality, execution time needed, and consistency of solution quality. As future work, we would like to optimize the parameters of our algorithm, for instance using meta-algorithms [1] and experiment with our approach in a distributed setting. Also, we plan to experiment with other algorithms, such as ant colony optimization or particle swarm optimization.

Acknowledgement

The authors are partially supported by the EU funded IST STREP Project FP6 - 26896: Time-Determined Ontology-Based Information System for Real Time Stock Market Analysis (http://www.semlab.nl/towl).

References

[1] W. A. de Landgraaf, A. E. Eiben, and V. Nannen. Parameter Calibration using Meta-Algorithms. In *IEEE Congress on Evolutionary Computation*, pages 71–78. IEEE Computer Society, 2007.

[2] F. Hogenboom, A. Hogenboom, R. van Gelder, V. Milea, F. Frasincar, and U. Kaymak. QMap: An RDF-Based Queryable World Map. In *Proceedings of the Third International Conference on Knowledge Management in Organizations (KMO 2008)*, pages 99–110, Vaasa, Finland, 2008.

[3] T. W. Manikas and J. T. Cain. Genetic Algorithms vs. Simulated Annealing: A Comparison of Approaches for Solving the Circuit Partitioning Problem. Technical report, University of Pittsburgh, 1996.

[4] M. Steinbrunn, G. Moerkotte, and A. Kemper. Heuristic and Randomized Optimization for the Join Ordering Problem. *The VLDB Journal*, 6(3):191–208, 1997.

[5] H. Stuckenschmidt, R. Vdovjak, J. Broekstra, and G. J. Houben. Towards Distributed Processing of RDF Path Queries. *International Journal of Web Engineering and Technology*, 2(2-3):207–230, 2005.