Application of SAPSO optimization algorithm to optimize SPARQL queries of BioMedical Datasets

Dr.R.Gomathi[1], Dr.S.Logeswari[2], Dr.B.Gomathy[3]

[1,2,3]Department of Computer science and Engineering,
Bannari Amman Institute of Technology Sathy, (India)

ABSTRACT

One of the emerging technologies in the recent years includes the Semantic Web technology. Semantic web stores information in much variety of formats. One of the most accepted formats is the Resource Description Framework (RDF) format. With an increase in the amount of semantic web data, querying large RDF graphs becomes a tedious process. Also the problem of query optimization becomes a concern in querying large RDF graphs. Choosing the best query plan reduces the amount of query execution time. To address this problem, nature inspired algorithms can be used as an alternative to the conventional query optimization techniques. In this research, the optimal query plan is generated by applying the SAPSO algorithm which is a hybrid of Simulated Annealing (SA) and Particle Swarm Optimization (PSO) algorithms. In this research, the SAPSO algorithm is applied to queries of biomedical datasets. The SAPSO algorithm finds the local positive solution and it avoids the problem of local minima. Testing was performed on biomedical queries with changeable number of predicates. The algorithm applied in this research gives better results compared to existing algorithms in terms of query execution time.

Keywords—Semantic web, RDF, Query optimization, Nature inspired algorithms, PSO, SA, biomedical datasets.

I. INTRODUCTION

Computers cannot easily recognize the data provided by the web pages but human beings can. The semantic web technology is a solution for the machines to interpret the information in the web pages. This technology allows computers to easily look for, merge and manipulate the information in the web. Information in the semantic web can be represented using a variety of formats like the RDF, Web Ontology Language (OWL), and Extensible Mark-up Language (XML). The most widespread of them is the RDF format. RDF is a data model to characterize information in the web. The syntax of XML and Uniform Resource Identifiers (URI) is used to identify resources. Each resource can be expressed using property and property values in RDF.

Nature has been growing for quite a lot of years, and it has been providing many inventive solutions to solve many problems. There are many sources of nature which embraces swarm intelligence, biological systems, physics based or chemistry based systems. A subset of metaheuristcs is often referred to as Swarm Intelligence (SI) based algorithms have been developed by imitating the so called swarm intelligence characteristics of biological agents such as birds, fish, humans, and many more. Examples include, Particle Swarm Optimization (PSO) based on the swarming behavior of birds and fish [1], the Firefly Algorithm (FA) based on the blinking pattern of tropical fireflies [2], and Cuckoo Search (CS) algorithm [3] inspired by the brooding parasitism of some cuckoo species. Physics and chemistry based algorithms include Harmony Search, Simulated Annealing [4] and so on. Nature inspired algorithms are applied for solving large scale problems. The language for querying data in the semantic web is the SPARQL protocol and RDF Query Language (SPARQL). This query language can be used to query data from varied sources, whether the data stored in native RDF format or it is viewed as RDF by means of any middleware. Biomedical datasets are larger in size and to query such data needs an algorithm to handle large scale problems. The paper is organized as follows: Section II reviews the methods used for query optimization in literature; Section III makes a study of the SAPSO algorithm; Section IV explains the application of the SAPSO algorithm to optimize the SPARQL queries of biomedical datasets; Section V describes the biomedical datasets used and the experimental results; Section VI sketches the conclusions and future works.

II. REVIEW OF RELATED WORKS

There is a requirement to optimize the join of the partial query results. An Ant Colony System (ACS) [5] was proposed in literature to query the semantic web setting effectively. The improvement in solution costs was compared with the existing algorithms like Genetic Algorithm (GA) and two phase optimization (2PO). It was shown that the ACS approach outperforms the existing approaches.

The basic idea connected with proficient processing [6] of SPARQL queries was studied in investigation. The study focus on i) equivalences of SPARQL algebra ii) the investigation of complexity analysis of all operators in SPARQL query language iii) optimizing SPARQL queries. The difficulty analysis showed that all fragments of SPARQL fall into the class of NP.

A new join ordering algorithm based on cardinality assessment [7] was proposed in writing for SPARQL query optimization. Experiments conducted on star queries and arbitrary queries illustrate that optimal query plans were found and executed in less amount of time.

To optimize a specific class of SPARQL queries called RDF chain queries, a new genetic algorithm was devised called the RCQ-GA[8]. The research work compares the performance of the algorithm with two phase optimization and the results reveal the superiority of the solution and consistency of the solution. The algorithm optimizes the chain queries by finding the order in which joins are to be carried out.

A parallel join algorithm was planned to handle RDF [9] and its query language SPARQL. The algorithm merge three join algorithms and it processes multiple queries in an interleaved approach. The performance results for a variety of data sets were discussed.

A new set of PSO algorithms to optimize queries in distributed databases [10] was offered in research. The particles representing solutions are moved with respect to a probability distribution. The algorithm looks for the near optimal quality execution plans. By varying the constraint setting of PSO, the execution times can be accustomed.

A language for querying graphs was projected in research [11] called the G-SPARQL. A hybrid query engine which partition the query plan and its parts are pushed inside the relational database and some parts are processed using memory based algorithms. The effectiveness and scalability of the proposed approach was discussed.

To reduce the query responding time, a cost model [12] using Map Reduce framework was explained to explore the scalability of RDF data. The search space is diminished by using a algorithm called All-Possible-Join tree (APJ-tree) algorithm. Tests were performed and the results were compared with the state of art solutions. A hybrid joins and a bloom filter to speed up the processing of joins was applied.

To evaluate SPARQL queries, a RDF query engine was designed in literature [13]. Experiments were conducted on both real and synthetic datasets to compute the performance. The evaluation method uses an index structure to index the RDF triples. A tree shaped optimization algorithm was designed to convert the SPARQL query graph into an optimal query plan.

III. OPTIMIZING BIOMEDICAL QUERIES BY APPLICATION OF THE SAPSO ALGORITHM

The following algorithm shows the [14] steps of the SAPSO algorithm for generating optimal query plan,

1. Initialize a population of N query plans and a Temperature value T. For the i^{th} query plan, its location $\boldsymbol{X}_i$ in the search space is randomly placed. Its velocity vector is $\boldsymbol{V}_i$= (v_{i1}, …, v_{id}… v_{iD}), in which the velocity in the d^{th} dimension is V_{id}=rand()×$\boldsymbol{V}_{\max}$, where rand() is the random number in the range [-1, 1].
2. Assign value for c1, c2, and w. pbest is the current best position of the particle and gbest is the current global best position of the particle.
3. Compare the evaluated fitness value of each query plan with its *pbest*. If the current value is better than *pbest*, then set the current location as the *pbest* location. Furthermore, if the current value is better than *gbest*, then reset *gbest* to the current index in the particle array.
4. Change the velocity of the particle using the following equation

$$V_{id} = V_{id} + c_1 \times rand() \times (P_{id} - X_{id}) + c_2 \times rand() \times (P_{id} - X_{id}) \quad (1)$$

5. Update the location of the particle and SA operator

$$X_{id} = X_{id} + V_{id} \qquad (2)$$

6. Reduce the Temperature value using the following equation

$$T = T * 0.95 \qquad (3)$$

7. Repeat Steps 3-5 until the number of iteration is greater than the allowable maximum iteration number $T_{\max}$ or till the optimum query plan achieved.

IV. EXPERIMENTAL ANALYSIS

The dataset used in this proposed research is the Bioportal dataset[15] to test the proposed algorithm. The bioportal is a dataset of linked biomedical ontologies and terminologies in RDF. This set includes ontologies that were developed in OWL, OBO and other formats, as well as a large number of medical terminologies that the US National Library of Medicine distributes in its own proprietary format. The RDF version of all these ontologies is published at http://sparql.bioontology.org. This dataset contains 190M triples, representing both metadata and content for the 300 ontologies. The proposed algorithm is tested in a Microsoft Windows XP platform on an Intel Pentium 4 machine with 2GB RAM. Biomedical queries are tested for performance of the algorithm. The number of predicates is varied and the proposed algorithm is iterated for 100 times. The execution time is recorded before and after application of the optimization algorithm.

Some sample queries include:

```
PREFIX meta: <http://bioportal.bioontology.org/metadata/def/>
 SELECT DISTINCT ?vrtID ?graph
WHERE {
    ?vrtID meta:hasVersion ?version .
    ?version meta:hasDataGraph ?graph .}
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX snomed-term: <http://purl.bioontology.org/ontology/SNOMEDCT/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT DISTINCT ?x ?label
FROM <http://bioportal.bioontology.org/ontologies/SNOMEDCT>
WHERE
{
    ?x rdfs:subClassOf snomed-term:363664003 .
    ?x skos:prefLabel  ?label.
}
```

The average execution time obtained for five different queries with varying number of predicates are recorded and compared with the query execution time before optimization. Figure 3.1 shows the average execution time in seconds.

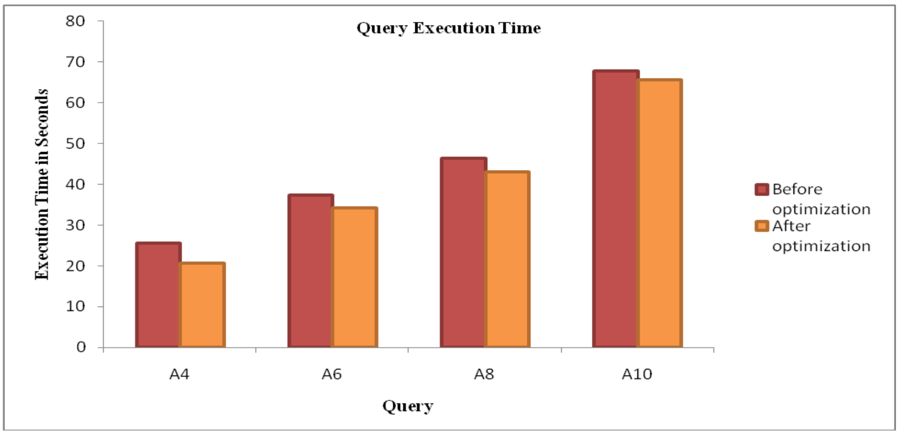

Figure 3.1 Query Execution Time

The experimental results shows that the query execution time gets reduced compared to querying without optimization. When the number of predicates increases also the query execution time gets reduced with the proposed algorithm.

V. CONCLUSION

In this work, the application of SAPSO algorithm to optimize biomedical queries is presented. The algorithm starts with a solution space consisting of all probable query plans. The fitness function for the optimization algorithm is defined which depends upon the cardinality and selectivity of the triples in the input biomedical dataset. The experimental results show the efficiency of the applied algorithm in terms of query execution time. The SAPSO algorithm is applied to biomedical queries having different number of predicates and the best plan is generated which in turn reduces the execution time of the query.

REFERENCES

[1] Kennedy, J, Particle swarm optimization, In Encyclopedia of Machine Learning, Springer US, 2010, 760-766.

[2] Yang, X. S, Nature-Inspired Metaheuristic Algorithms, Luniver Press,2008.

[3] Yang, XS& Deb, S,Cuckoo search via Levy flights, Proceedings of IEEE World Congress on Nature and Biologically Inspired Computing, 2009,210-214.

[4] Kirkpatrick, S& Vecchi, MP 1983, ’Optimization by simulated annealing’, Science,220(4598),671-680.

[5] Hogenboom, A, Niewenhuijse, E, Hogenboom, F& Frasincar, F , RCQ-ACS: RDF Chain Query Optimization Using an Ant Colony System, Proceedings of the international conferences on web intelligence and Intelligent Agent Technology, 1,2012,74-81.

[6] Schmidt, M, Meier, M& Lausen, G,Foundations of SPARQL query optimization, Proceedings of the thirteenth international conference on database theory, 2010, 4-33.

[7] Vidal, ME, Ruckhaus, E, Lampo, T, Martínez, A, Sierra, J& Polleres, A, Efficiently joining group patterns in SPARQL queries, In The Semantic Web: Research and Applications, Springer Berlin Heidelberg, 2010, 228-242.

[8] Hogenboom, A, Milea, V, Frasincar, F& Kaymak, U ,RCQ-GA: RDF chain query optimization using genetic algorithms, Springer Berlin Heidelberg, 2009, 181-192.

[9] Senn, J, Parallel Join Processing on Graphics Processors for the Resource Description Framework, Proceedings of the twenty-third international conference on architecture of computing systems, 2010, 1-8.

[10] Dokeroglu, T, Tosun, U& Cosar, A, Particle Swarm Intelligence as a new heuristic for the optimization of distributed database queries, Proceedings of the sixth international conference on Application of Information and Communication Technologies, 2012, 1-7.

[11] Sakr, S, Elnikety, S, & He, Y , Hybrid query execution engine for large attributed graphs, Information Systems,.41,2014, 45-73.

[12] Zhang, X, Chen, L& Wang, M , Towards efficient join processing over large RDF graph using mapreduce, Scientific and Statistical Database Management, Springer Berlin Heidelberg, 2014, 250-259.

[13] Liu, C, Wang, H, Yu, Y& Xu, L ,Towards efficient SPARQL query processing on RDF data, Tsinghua Science & Technology, 15(6),2010, 613-622.

[14] Gomathi, R& Sharmila, D , A Hybrid Nature Inspired algorithm for generating optimal query plan, World Academy of Science Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 8(8),2014,1433-1438.

[15] Manuel Salvadores, Paul R. Alexander, Mark A. Musen, and Natalya F. Noy' BioPortal as a Dataset of Linked Biomedical Ontologies and Terminologies in RDF, Semantic Web Journal,.4(3),277-284.