Mathematical Model of Semantic Look - An Efficient Context Driven Search Engine

Leena Giri G[a], Srikanth P L[a], Manjula S H[a], K R Venugopal [a], L M Patnaik[b]

[a]Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore 560 001 India, Contact: leenagiri@gmail.com.

[b]Honorary Professor, IISc., Bangalore.

The World Wide Web (WWW) is a huge conservatory of web pages. Search Engines are key applications that fetch web pages for the user query. In the current generation web architecture, search engines treat keywords provided by the user as isolated keywords without considering the context of the user query. This results in a lot of unrelated pages or links being displayed to the user. Semantic Web is based on the current web with a revised framework to display a more precise result set as response to a user query. The current web pages need to be annotated by finding relevant meta data to be added to each of them, so that they become useful to Semantic Web search engines. Semantic Look explores the context of user query by processing the Semantic information recorded in the web pages. It is compared with an existing algorithm called OntoLook and it is shown that Semantic Look is a better optimized search engine by being more than twice as fast as OntoLook.

Keywords : Ontology, RDF, Semantic Web.

1. INTRODUCTION

Semantic Web (Web 3.0) is the proliferation of unstructured documents of the web to a "web of data" [1]. In traditional web architecture there is less emphasis on meta data of the web document during the data collection phase of the search engine and the concentration is more on classic approaches like Information Retrieval and Natural Language Processing. It is difficult to know the context or the role played by the web document designed for such approaches [2][3][4]. This is overcome by Semantic Web where enhanced version of meta data are embedded in the web pages as RDF [5] and Ontology [6]. Ontology defines the concepts and the relations between these concepts. RDF (Resource Description Framework) describes the web document in the form of triplets. Every RDF triplet is a composition of *subject*, *predicate* and *object*. *Subject* is an entity to be described, *object* is an entity which describes the subject and *predicate* is a relationship between *subject* and *object*; essentially every *predicate* describes the different context of the web page playing multiple roles. Both Ontologies and RDF are embedded in web pages forming the semantic annotation of a web page.

1.1. Motivation

The existing search engines interpret the keywords of a user query in isolation without considering wholly, the context of the search query. Because of this, most of the results retrieved is irrelevant to the user query. This hits the performance and accuracy of search engines. The main purpose of providing multiple keywords is to make search based on a particular context. It is to say that nothing exists without context or relation. As an example, consider a scenario where a user has submitted the keywords "Ashoka+Bangalore+Hotel" with the intention to search for Hotel Ashoka in Bangalore. Traditional web search engines return all the web pages containing the keywords "Ashoka, Bangalore and Hotel" without considering the context of the user query. Most of the web pages are irrelevant to the user query; where some pages may provide information on "Ashoka Pillar in Bangalore", some web pages may provide information on different hotels because of the keyword 'hotel', and some pages may provide information about Bangalore city because of the keyword 'Bangalore'.

1.2. Contribution

Semantic Look processes the semantic annotation embedded in web pages to perform the relevance analysis on the user query to retrieve the related URLs. It constructs the Ontology graphs which are eqvivalent to the Concept-Relation graph of OntoLook [7]. Semantic Look is optimized compared to OntoLook, where heavy weighted arcs are retained in every sub graph and only half of the number of least weighted arcs are pruned which mitigates the number of sub graphs to be processed.

1.3. Organization

This Section describes the rest of the paper in brief. Section 2 describes a few earlier contributions to the implementation of Semantic Web based Search Engines. Section 3 defines the problem considered. Section 4 explains the architecture of Semantic Look. In Section 5 the mathematical model for Semantic Look is given by describing the notations used and the equations that result in a more accurate result set being displayed. Section 6 describes the algorithms required in the components of the system architecture. Section 7 gives the experimental results mentioning the data sets considered and a graphical representation of performance evaluation between OntoLook and Semantic Look. The paper concludes by mentioning the enhancement that can be incorporated in Semantic Look and the list of references considered by the authors.

2. RELATED WORK

Traditional search engines return keyword-isolated pages because of which many unrelated pages or web links are returned by them in response to the user's query. Semantic search engines can be categorized into different types of which context based search engine is one. This is the largest group and the aim here is to add semantic operations for better search results. *Yufei Li et. al.,* [7] have proposed a prototype called OntoLook for performing relation based search to derive the context of user query. Concept-Relation Graphs (CRGs) are created with vertices representing concepts and edges representing the number of relations between these concepts. An algorithm generates sub graphs by pruning edges of the CRG irrespective of whether they are heavy weighted or less heavy weighted edges and this results in large number of sub graphs to be processed.

Junghoo Cho, Hector Garcia-Molina, Lawrence Page [8] have proposed the theory based on "Efficient Crawling Through URL Ordering", in order to obtain more "important" pages first. Obtaining important pages rapidly can be very useful when a crawler cannot visit the entire web in a reasonable amount of time. They define several important metrics, ordering schemes, and performance evaluation measures for this problem. They experimentally evaluate the ordering schemes on the Stanford University Web to prove that a crawler with a good ordering scheme can obtain important pages significantly faster than one without agood ordering scheme.

Li Ding et. al., [9] have proposed the theory based on "Search on the Semantic Web". They propose this theory based on the fact that, in order to help human users and software agents find relevant knowledge on the Semantic Web, Swoogle, a search engine, discovers, indexes and analyzes the Ontologies and facts that are encoded in Semantic Web documents. *Natalya F Noy et. al.,* [10] have proposed the theory on creating "Semantic Web Contents". As researchers continue to create new languages in the hope of developing a Semantic Web, they still lack consensus on a standard. They describe how Protege-2000 - a tool for Ontology development and knowledge acquisition - can be adapted for editing models in different Semantic Web languages. Semantic Web is necessary to express information in a precise, machine interpretable form, so software agents processing the same set of data share an understanding of what the terms describing the data mean.

Lastra J L M et. al., [11] study the use of Semantic Web Services in order to overcome this challenge. The use of Ontologies and explicit semantics enable performing logical reasoning to infer sufficient knowledge on the classification of processes that machines offer, and on how to execute and compose those processes to carry out manufacturing orchestration autonomously. *Yi Jin* [12] present an architecture of the Semantic Search Engine and our work shows how the fundamental elements of the Semantic Search engine can be used in the fundamental task of information retrieval. An improved version of the tf-idf (term frequency -inverse document frequency) algorithm is proposed to guarantee the retrieval of information resources in a more efficient by looking for items in which the the keywords that are searched for are more common than usual.

Alexander Maedche et. al., [13] have presented an integrated enterprise-knowledge management architecture for implementing an Ontology based Knowledge Management System (OKMS) and have made a study on two critical issues related to working with Ontologies in real-world enterprise applications. *Wang Young-gui et. al.,* [14] have done analysis on application of Semantic Web to web mining and to build a semantic based web mining model under the framework of the Agent. The authors in [15] discuss clustering as a method to overcome the problem of searching through the list a search engine displays. The list that is displayed is extremely inconvenient to the users since it expects them to look into each page sequentially in an exhaustive manner which could result in relevant information being overlooked. *Mohammad Farhan Husain et. al.,* [16] discuss how current frameworks do not scale for storing large RDF graphs and describe a framework that is built using Hadoop by exploiting the cloud computing paradigm.

3. PROBLEM DEFINITION

Given a set of keywords for a search, the main goal is to find the set of web pages related to the user search context by extracting the semantics behind the user query, where the result set contains most relevant web pages with unnecessary pages filtered from it. It is assumed that every web page consists of embedded RDF (e-RDF) and embedded Ontology (e-Ontology) forming the semantic annotation for a web page. The web pages with e-RDF and e-Ontology form the Semantic World Wide Web which is used by the crawler for crawling the e-RDF/e-Ontology in the web pages.

Semantic Look, a variant of OntoLook, optimizes this search logic by retaining the high ranked edges in every sub graph, as they are relevant to the user query, and pruning only the least weighted arcs. As an example, if the number of edges in the CRG is 7 of which there are 4 less weighted arcs, OntoLook produces 2^7 *i.e.*, 128 sub graphs. Semantic Look produces only 6 sub graphs, since it prunes half the number of least weighted arcs *i.e.*, ${}^4C_2$.

4. SYSTEM ARCHITECTURE

The Context Driven Search Engine includes Semantic Crawler, Semantic Parser, Semantic Look and Ontobase as components for drawing the context of user search, as shown in Fig. 1. The context is drawn for a user query by extracting the relations among the keywords submitted by the user. The relations are recorded in RDF adhering to a particular Ontology, like the travel Ontology, and is embedded in every web page as semantic meta data. Semantic Web uses these meta data as information for searching the web pages.

4.1. Semantic Crawler

The Semantic Crawler collects the e-RDF/e-Ontology present in the web pages and invokes the corresponding parsers depending on the document type. Both RDF and Ontology are serialized as XML and embedded in a web page forming semantic meta data for a web page. The crawler performs the collection of web page contents and maps it to a web page database.

4.2. Semantic Parser

This component of the search engine is responsible for parsing the incoming Semantic Annotation sent by the crawler. RDF parser parses the e-RDF documents to generate RDF-triplets

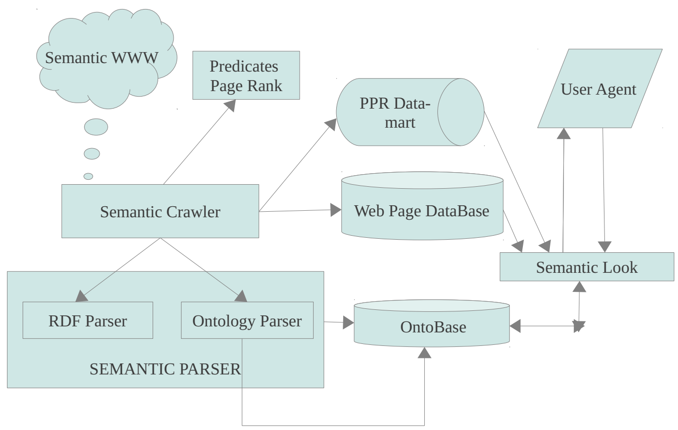


Figure 1. System Architecture

and parses the e-Ontology to generate Ontology triplets. The generated triplets are mapped to Ontobase which is a knowledge base containing semantic information.

4.3. Semantic Look

The core component of the search engine is the Semantic Look. This utilizes the semantic information to develop all possible contexts for the user query. The context can be developed by extracting the relations between the keywords which are obtained by the corresponding Ontology and RDF triplets. The RDF triplets generated for the user query is used to fetch URL set.

The Semantic Look

- captures keywords and its corresponding concepts to form an Ontology graph where vertices represent concepts and edges represent relations between these concepts. Integers on the edges represent the number of relations between the concept pairs.

- forms the less ranked arcs set and decide on the number of arcs to cut, by considering the average number of relations between the arcs.

- cuts only the less ranked arcs to form the Ontology sub graph that will optimize the search result

- processes the Ontology subgraph to produce all possible Ontology triplets from Ontobase

- uses the Ontology triplets to form all possible RDF triplets for the user query

- submits the RDF triplets to Ontobase to fetch URL set and sort according to the ranks assigned

The sequence of execution is as shown in Fig. 2.

4.4. Ontobase

Ontobase is a knowledge base containing semantic annotation of web pages which includes both Ontology and RDF triplets. Semantic Look uses this semantic annotation to obtain all possible RDF triplets defining the context of the user query.

5. MATHEMATICAL MODEL

This section describes the mathematical model of Semantic Look using mathematical concepts and language. The model helps in explaining

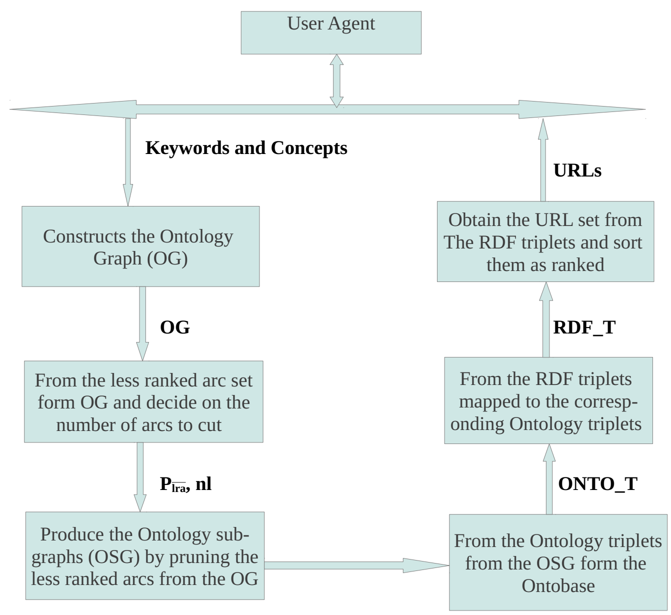


Figure 2. Functional Diagram

the system and to study the effects of different components, to analyse and predict its behaviour. Table 1 mentions the notations used by the authors and the relevant meanings of the notations used.

5.1. Definitions

Webpage keywords: It is the set of keywords vectors accessed according to web page index. It is given by KW=$\{KW_1$, $KW_2 ... KW_{|}web|\}$. Any i^{th} vector is defined as

$$KW_i = \{k_l \mid 1 \leq l \leq |KW_i\,|\} \tag{1}$$

where KW_i is the i^{th} web page keyword and k_l is the l^{th} keyword in KW_i.
Web page Concepts: It is the set of concepts vectors accessed according to web page index. It is given by CW=$\{CW_1$, $CW_2 ... CW_{|}web|\ \}$. Any i^{th} vector is defined as

$$CW_i = \{c_l \mid 1 \leq l \leq |CW_i\,|\} \tag{2}$$

where CW_i is the i^{th} vector of web page concepts mapped to the i^{th} keyword vector, *i.e.*, KW_i of web page keywords and c_l is the l^{th} concept in CW_i.
Web page Predicates: It is the set of relation vectors specifying the relations between the concept pairs in CW and accessed according to webpage index. It is given by RW $\leftarrow$$\{RW_1$, RW_2,...,$RW_{|}web|\}$. Any i^{th} vector is defined as

$$RW_i = \{r_{lk}, r_{kl} \mid 1 \leq l, k \leq C_2^{|CW_i|, l \neq k}\} \tag{3}$$

where RW_i is the i^{th}vector of web page predicate with the domain and range concepts interchanged.
Ontology Triplets of Web Page: The Ontology triplet consists of subject concept called domain and object concept called range with predicate/relationship specyfying the relation between subject and object concepts. The Ontology triplets is a set on Ontology triplets vectors defined as $ONTO\_T$=$\{ONTO\_T_1$, $ONTO\_T_2$,...,$ONTO\_T_{|}web|\}$. Any i^{th} vector

Table 1
Basic Notations

Notations	Meaning
K	Set of keywords submitted by the user.
C	Set of concepts mapped for the keywords given by the user
λ	Threshold: Minimum support count for the occurences of relations in a Web Page
web	Set of URLs of web pages accessed according to Web page index
RDF_T	Set of RDF triplets accessed according to web page index
ONTO_T	Set of Ontology triplets indexed by Webpage index
R	Set of relation vectors between the corresponding concept pairs given by the user
OG	2-D matrix representing Concept-Relation graph
N	Total number of arcs in Ontology graph
nl	Total number of less ranked arcs where nl<n
p(r,w)	Probability of occurrence of relation r, R(w) in a web page w. WI is referred to as *predicate frequency*
KW	Set of keyword vectors accessed according to web page index
CW	Set of Concepts vectors accessed according to web page index
R	Set of Relations vector between the concept pairs
RW	Set of Relations vector

is defined as

$$ONTO\_T_i = \left\{ \begin{array}{l} (CW_{id}, CW_{ij}, CW_{ir}) \\ (CW_{ir}, CW_{ij+1}, CW_{id}) \mid \\ 1 \leq d, r \leq C_2 \ |^{CW_i}| \\ and\ 1 \leq j \leq \mid RW_i \mid \\ and\ d \neq r\} \end{array} \right\} \quad (4)$$

where CW_{id},CW_{ir} are the domain and range concepts and RW_{ij} is the j^{th} relation in the i^{th} web page specifying the relation between CW_{id} and CW_{ir}. Similarly RW_{ij+1} is a relation between CW_{id} and CW_{ir} with the domain and range concepts interchanged.

RDF Triplets of Web Page: The RDF triplet consists of subject and object with predicate/relationship specifying the relation between corresponding subject and object concepts. The RDF triplets is a set on RDF triplets vectors defined as RDF_T=\{RDF_T_1, RDF_T_2,...,$RDF\_T_{|}web|$\}. Any i^{th} vector is defined as

$$RDF\_T_i = \left\{ \begin{array}{l} (KW_{id}, RW_{ij}, KW_{ir}) \\ (KW_{ir}, RW_{ij+1}, KW_{id}) \mid \\ 1 \leq d, r \leq C_2 \ |^{KW_i}| \\ and\ 1 \leq j \leq \mid RW_i \mid \\ and\ d \neq r\} \end{array} \right\} \quad (5)$$

where KW_{id}, KW_{ir} are the domain and range concepts and RW_{ij} is the j^{th} relation in the i^{th} web page specifying the relation between KW_{id} and KW_{ir}. Similarly RW_{ij+1} is a relation between KW_{id} and KW_{ir} with the domain and range concepts interchanged.

Ontology Graph: The concepts received from the user are paired to form Ontology graph, where the vertices represent the concepts and the edges represent the number of relationships existing between the concept pairs. The Ontology graph is represented as 2-D matrix where the value in i^{th} row and j^{th} column is defined as:

$$OG[i,j] = \left\{ \begin{array}{l} \infty,\ if\ i \neq j\ and \mid R_{ij} \mid = 0 \\ 0 < \mid R_{ij} \mid < \infty,\ if\ i \neq j \\ and \quad \mid R_{ij} \mid > 0 \\ 0,\ if\ i = j \end{array} \right\} \quad (6)$$

where $R_{ij} \in R$ is a vector representing all the possible relations between the concept pairs C_i

and C_j and is defined as:

$$R_{ij} = \left\{ RW_{wk} \mid \begin{array}{l} 1 \leq w \leq \mid \ RW_w \mid \\ 1 \leq d, r \leq C_2^{|CW_i|}, \\ 1 \leq k \leq \mid RW_w \mid \\ and\ CW_{wd} = C_i \\ and CW_{wr} = C_j \end{array} \right\} \tag{7}$$

6. MODEL for SEMANTIC LOOK

Semantic Look retains the high weighted arcs in the graph and prunes only the less weighted arcs to produce sub graphs from Ontology graph to be processed. The results are, therefore, relevant to the user query. The RDF triplets generated from the result of sub graph processing is submitted to web page database to fetch URL set.

Theorem 1:"The search engine time is reduced by pruning the less ranked arcs and the results are more relevant to user query".

Proof:

Let

$$lra = min(OG) \tag{8}$$

Find $l$ra which is the arc with the less weight. min(OG) returns minimum edge from Ontology graph.

$$\overrightarrow{P_{lra}} = \{e_{ij} \mid 1 \leq i \leq OG \mid, lra = OG[i,j]\} \tag{9}$$

where $\overrightarrow{P_{lra}}$ is a set of less ranked arcs and therefore $nl = \mid \overrightarrow{P_{lra}} \mid$ where nl indicates number of such less ranked arcs and $nc = \lceil \frac{nl}{2} \rceil$ where nc indicates the number of less ranked arcs to be cut in Semantic Look. The average number of arcs are pruned from the Ontology graph. Since $nl < N$ and the total number of sub graphs which are candidates for processing are $2^{nl} < 2^N$ and actual number of sub graphs processed are $\binom{2^{nl}}{nc} < \binom{2^N}{nc}$. Since high ranked arcs are retained in every sub graph, the result is more relevant to the user query and the search time is reduced by pruning only the less ranked arcs which produces 2^{nl} sub graphs which is less than 2^N sub graphs produced by not only pruning the less ranked arcs but also high ranked arcs from the Ontology graph.

6.1. Implementation

The proposed search engine called Semantic Look which is a variant of ONTOLOOK [1] optimizes the search engine time by pruning the less ranked arcs from the Ontology graph and retaining the high ranked arcs in every sub graph, which mitigates the number of sub graphs to be processed by the search engine. The *Semantic Crawler* collects the semantic annotations embedded in every web page which includes embedded RDF and its corresponding Ontology (e-RDF/e-Ontology). The collected semantics and web page contents are stored in a database which is used by Semantic Look to perform the search for the user query. The *Semantic Parser* encapsulates the logic of the Ontology and RDF triplets to the database.

Table 2
Algorithm for Semantic Crawler

Input: Semantic World Wide Web
Output: Web pages, e-RDF and e-Ontology
Process:
$w$eb = Semantic World Wide Web
for each webpage w Web
do
  Li=search<link>with type=
    application/rdf+xml;
for each $l \in$ Li
  do
    O = new Semantica Parser();
    Href = href of <link>
    $u$rl = url of w
    $r$ootTag = parse the webpage whose
    url is set in href of
    <link> to fetch the root element.
if(rootTag =<ONTOLOGY>)
  then
    $O \rightarrow O$ntologyParser(href,url);
    else
      $O \rightarrow R$DFParser(href,url);
  done
Store the contents of w in database.
done

The *Semantic Crawler* crawls the web documents from the *Semantic World Wide Web*

Table 3
Algorithm for Ontology Parser

```
Input: Ontology document URL (ourl),
URL of Web Page(wurl)
Output: Ontology Triplets mapped to
the database
Process:
Step 1:
i = 0;
for each ObjectProperty as op
and ++i != ObjectProperty.length  do
if op.hasAttribute(rdf:ID)
  then
    Relation = sp.getAttribute(rdf:ID);
for each op.childNodes as ch
do
    if ch.nodeName = domain
      domain =ch.getAttribute(rdfs:resource);
    else
      range = ch.getAttribute(rdfs:resource);
    end if
done
end if
create
ONTO_T_wurli = (Domain, Relation, Range);
ONTO_T_wurli = ONTO_T_wurl ∪
                 ONTO_T_wurli;
done
ONTO_T = ONTO_T ∪ ONTO_T_wurli;
repeat the above process for Data
TypeProperty and FunctionalProperty
insert Ontology triplet to database.
```

Table 4
Algorithm for RDF Parser

```
Input: RDF document URL (ourl),
URL of Web Page (wurl)
Output: RDF_TRIPLETS of wurl
i.e. RDF_Twurl
Process:
i = 0;
  for each rdf:Description as d and
    ++i!=rdf:description.length
    do
      Subject = d.getAttribute(rdf:about);
      for each d.childNodes as c
        do
        Relation=c.nodeName;
        Object = c.getAttribute(rdf:resource);
        done
      end for
create RDF_T-wurl U RDF_T_wurli
  end for
obtain corresponding ONTO_T_wurl
insert ONTO_T_wurl and RDF_T_wurl
to the database.
```

Semantic – WWW and invokes the *Ontology parser* if the collected document is the Ontology document or invokes the RDF-parser if the collected document is the RDF document. The crawler also stores the web pages in the database for future use as explained in the algorithm given in Table 2. The Ontology parser creates the Ontology triplets by fetching the domain and range concepts of *ObjectProperty, DatatypeProperty* or *FunctionalProperty* and maps the corresponding triplets to the database as given in the algorithm of Table 3.

The RDF parser looks for the *<rdf:description>* element for forming the RDF-triplet and maps it to the corresponding Ontology triplets before it is added to the database. The *subjects/objects* are the instances of some Ontology concepts defined in the Ontology document which are described under the element called *<instances>* in RDF-document. The RDF-parser uses this information to map RDF-triplets to the Ontology triplets as explained in the algorithm of Table 4. *Semantic Look* dynamically constructs the Ontology graph from the concepts and corresponding keywords given by the user. The search engine cuts the less weighted arcs from the graph retaining high ranked arcs. There is no set criterion for deciding to cut such number of arcs but in this search engine average number of arcs are pruned from the Ontology graph to form the sub graphs. Since high ranked arcs are retained in every sub graph, the results obtained are relevant to the user query. The sub graphs are processed to from all possible RDF-triplets which are submitted to the database to retrieve URL set. Since there is a probability that RDF-triplets are repeated in multiple web pages the final URL set is obtained by the intersection of URL sets of all RDF-triplets matching the user query as explained in the algorithm of Table 5.

```
/*Generate RDF Triplets to fetch URLs*/
Procedure Generate_RDF_T(OSG)
do
URL = φ
for i,j <| OSG |
do
R_ij = RW_wk if
( 1 <= w <=| RW_w | and
    1 <= d, r <= C_2^{|CW_i|}, and
    1 <= k <=| RW_w |
     and KW_wd = K_i, KW_wr = K_j )
for k <| R_ij |
    do
      urli = {Web_i | (k_i, R_ijk, k_j)
        ∈ RDF_T_i}
      URL = URL ∩ urli
    done
done
rsort(url);
done
```

7. EXPERIMENTAL RESULTS

The dataset consists of 40 web pages with embedded 370 RDF and 85 Ontology triplets. The web pages are mapped to the web page database, where RDF and Ontology triplets are mapped to Ontobase. The RDF documents, Ontology documents and web pages form the Semantic World Wide Web. The XML serialized Ontology and RDF triplets are mapped to the tables in Ontobase which has complete information about the triplets such as subject, subjects concept ,object, objects concept, predicates and predicates type. The concept and predicates type are obtained from the Ontology document. Semantic Look is simulated on the context of the tourism portals. The search engine is generic but the e-Ontology and e-RDF is domain specific. The Semantic

Table 5
Algorithm for Semantic Look

```
Input: Key words set i.e. K,
Concepts set i.e. C
Data Store: Semantic World Wide Web
i.e. Web
Output: URL Set
Process:
Cn =| C |
lra = ∞
let R = R12, R21, R23, R32, . . . , Rcncn-1
  for each i,j < cn and i ≠ j do
      OG[i, j] =| R_ij |
        if OG[i, j] < lra then
           lra = OG[i,j]
        end if
    done
P⃗_lra = Φ
  for each i, j < cn and i!=j do
      if OG[i, j] = lra then
         P⃗_lra = P⃗_lra U e_ij
      end if     done nl =| P⃗_lra |; ceil(nl/2);
i =0;
Pstack;
While i < nl do
    k=i
    .push(P⃗_lrak)
  while !Pstack.empty() do
      if Pstack.length = nc;
      OSG[ ];
  for each Pstack[j] as e_lm do
      copy OG[ ] to OSG[ ]
      [l,m] = OSG[m,l] = ∞
      Generate_RDF_T(OSG)
    done
  Pstack.pop()
  end if
else
do
    if ++k≠nl then
      .push(P⃗_lrak)
    end if
k = P⃗_lra.position(Pstack.top)
Pstack.pop()
done
done
done
```

Crawler crawls the e-Ontology and e-RDF without any knowledge on the context where these documents are playing a role. The search context for a user query is established by extracting the relations among the supplied keyword. This is performed by *Semantic Look*.

The entire application is developed on *LAMPP* environment with*PHP* as underlying language for business logic. As shown in *Fig 3* the Semantic Look and Ontolook is compared with respect to the number of relations to be processed for different sets of keywords and concepts provided by the user.The difference in the number of sub graphs processed by OntoLook and Semantic Look is given in Table 7.

Since in every sub graph high ranked edges are retained and only the selected less ranked edges are pruned, the number of sub graphs to be processed is less in Semantic Look compared to Ontolook. As shown in *Table 7*, the number of relations to be processed in Semantic Look is less than half of the number of relations processed by Ontolook as depicted in *Figure 3*. Every sub

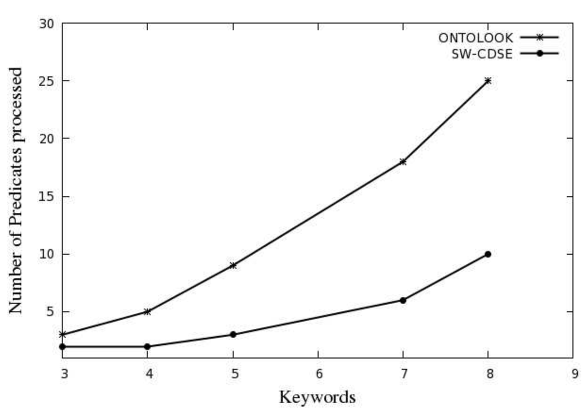


Figure 3. Keywords Predicates Processed

graph produces large number of duplicate RDF triplets which is submitted to the Ontobase to fetch URLs for every sub graph and intersection of these URL sets produce the distinct set of URLs as a result set for the user. The search time here includes the time for pruning the selected less ranked edges from the Ontology graph, producing the RDF triplets and database communication time for fetching the URLs set from it. From *Table 7* and *Figure 3* it is shown that number of sub graphs produced in Semantic Look is less compared to Ontolook and therefore the number of RDF triplets produced in Semantic Look is less which in turn reduces the search time as compared with Ontolook. *Table 8* shows the number of RDF triplets processed and search time taken by Ontolook and Semantic Look.

8. CONCLUSION

Search engines in the current web architecture will not consider the semantics role played by web pages in different context. The new generation of web i.e. Semantic Web (web 3.0) considers this context information by recording the semantic information in the form of Ontologies and RDFs. A proof of concept called Semantic Look is proposed to produce relevant web pages by filtering unnecessary web documents from the result set.

Semantic Look extracts the semantics of the user query to know the context of user search. This work is based on the prototype called OntoLook which performs the exhaustive search of all the sub graphs of Ontology graph to produce URL set. Semantic Look is an optimized search engine compared to OntoLook which prunes less weighted edges from the OntoLook to produce less number of sub graphs for processing.

Even though the number of sub graphs processed by Semantic Look is less as compared with OntoLook the number of RDF triplets produced will be huge and therefore in future work Semantic Look should be designed to run on the clusters of nodes using Map-Reduce Framework. Further optimization is achieved by running the crawler and pruning logic on the cluster. Since semantic information is embedded in the web page by the author and it is assumed to be true there is a chance of misleading the search engine by embedding false semantic information.

Table 6
Sub Graphs Processed for a Particular Combination Keywords and Relations

No.of Keywords		No.of Relations		No.of Sgraphs processed	
OLook	SLook	OLook	SLook	OLook	SLook
8	8	25	10	5200300	252
7	7	18	6	48620	20
5	5	9	3	26	3
4	4	5	2	10	2
3	3	3	2	3	2

Table 7
No. of RDF Triplets Produced and Search Time to process them for Combination Keywords and Relations

Sub graphs processed		RDF triplets produced		Process Time	
OLook	SLook	OLook	SLook	OLook	SLook
5200300	252	701345778	81144	710039	34.4076
48620	20	5209920	4320	21.0912	1.874
126	3	6832	354	3.2049	0.216094
10	2	244	120	0.12911	0.0634
3	2	48	46	0.02599	.02399

REFERENCES

1. W3 Consortium, Semantic Web, *http:// en.wikipedia.org/wiki/Semantic Web*.
2. T Priebe, C Schliiger and G Pernul. A Search Engine for RDF Metadata, *Proceedings of the* 15^{th} *International Workshop Database and Expert Systems Applications*, pages 168-172, 2004.
3. T Berners-Lee, J Hendler and O Lassila. The Semantic Web, *Scientific American*, 284(5):34-43, 2001.
4. A Gomez-Perez and O Corcho. Ontology Languages for the Semantic Web, *IEEE Intelligent Systems*, 17(1):54-60, January-February 2002.
5. D Beckett. RDF/XML Syntax Specification (Revised), *http://www.w3.org/TR/2004 /REC-rdf-syntax-grammar-20040210/, 1994*.
6. John Hebler, Matthew Fisher, Ryan Blac and Andrew Perez-Lopez. Semantic-Web Programming, *Wiley India Pvt.ltd*, Third Edition, 2009.
7. Yufei Li, Yuan Wang and Xiaotao Huang. A Relation-Based Search Engine in Semantic Web, *IEEE Transactions on Knowledge and Data Engineering*, 19(2):273-281, February 2007.
8. J Cho, H Gareia-Molina, and L Page. Efficient Crawling through URL Ordering, *Proceedings of the* 8^{th} *International World Wide Web Conference*, May 1999.
9. Li Ding, Finin, Joshi, Pan, Scott Cost, Sachs, Doshi, Reddivari, and Y. Peng. Swoogle, a Search and Metadata Engine for the Semantic Web, *Proceedings,* 13^{th} *ACM Conference on Information and Knowledge Management (CIKM 2004)*, Nov. 2004.
10. Noy, Sintek, Decker, Crubezy, Fergerson, and M A Musen. Creating Semantic Web Contents with Protege-2000, *IEEE Intelligent Systems*, 16(2):60-71, March-April 2001.
11. Lastra L M J and Delamer M. Semantic web Services in Factory Automation: Fundamental Insights and Research Roadmap, *IEEE Intelligent Systems*, 6(6):1-11, February 2006.
12. Yi, Jin, Zhuying Lin and Hongwei Lin. The Research of Search Engine Based on Semantic Web, *IEEE Intelligent Systems*, 360-363, December 2008
13. Alexander Maedche, Boris Motik, Ljiljana Stojanovic, Rudi Studer, and Raphael Volz. Ontologies for Enterprise Knowledge Management, *IEEE Computer society*, 2003.
14. Wang Young-gui and Jia Zhen. Research on Semantic Web Mining, *IEEE conference Publications*, 1:67-70, June 2010.
15. Ramesh Singh, Dhruv Dingra and Aman Arora. SCHISM - A Web search engine using seman-

tic taxonomy, *IEEE POTENTIALS*, September-October 2010.

16. Mohammad Farhan Husain, James McGlothlin, Mohammad Mehedy Masud, Latifur R. Khan and Bhavani Thuraisingham. Heuristics-Based Query Processing for Large RDF Graphs using Cloud Computing, *IEEE Transactions on Knowledge and Data Engineering*, 23(9), September 2011.

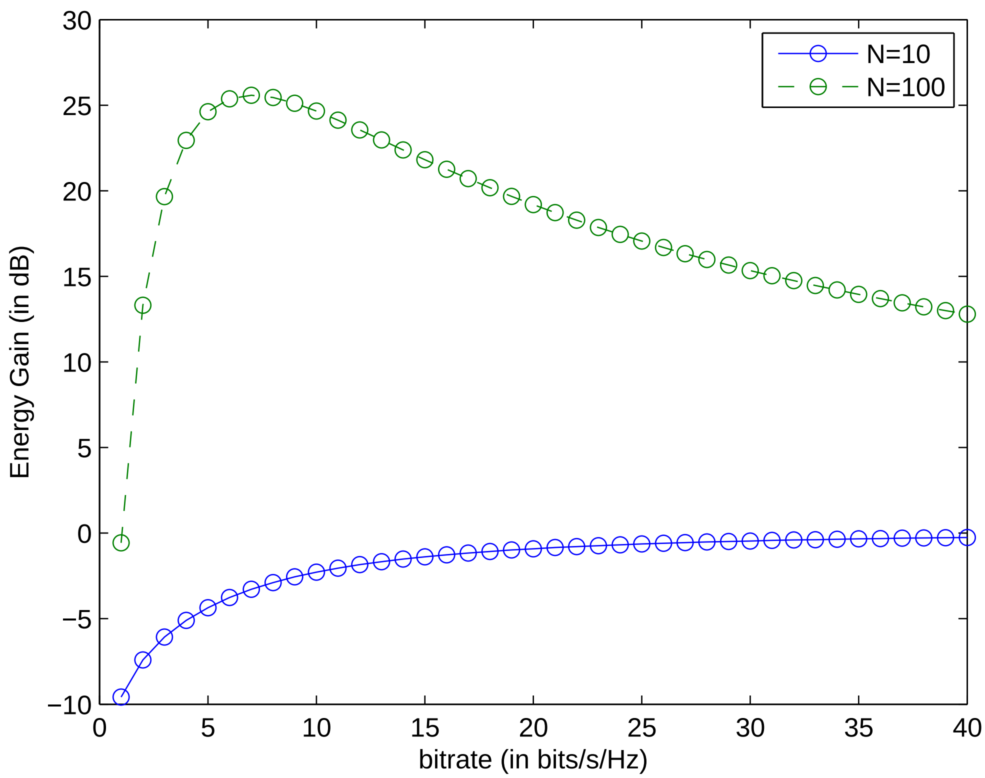
Leena Giri G is currently an Associate Professor in the Department of Computer Science, Dr. Ambedkar Institute of Technology, Bangalore. She obtained her Bachelor of Engineering from SJCE, Mysore. She received her M.Tech Degree in Computer Science and Engineering from IIT Mumbai. Her research interest is in the area of Semantic Web.

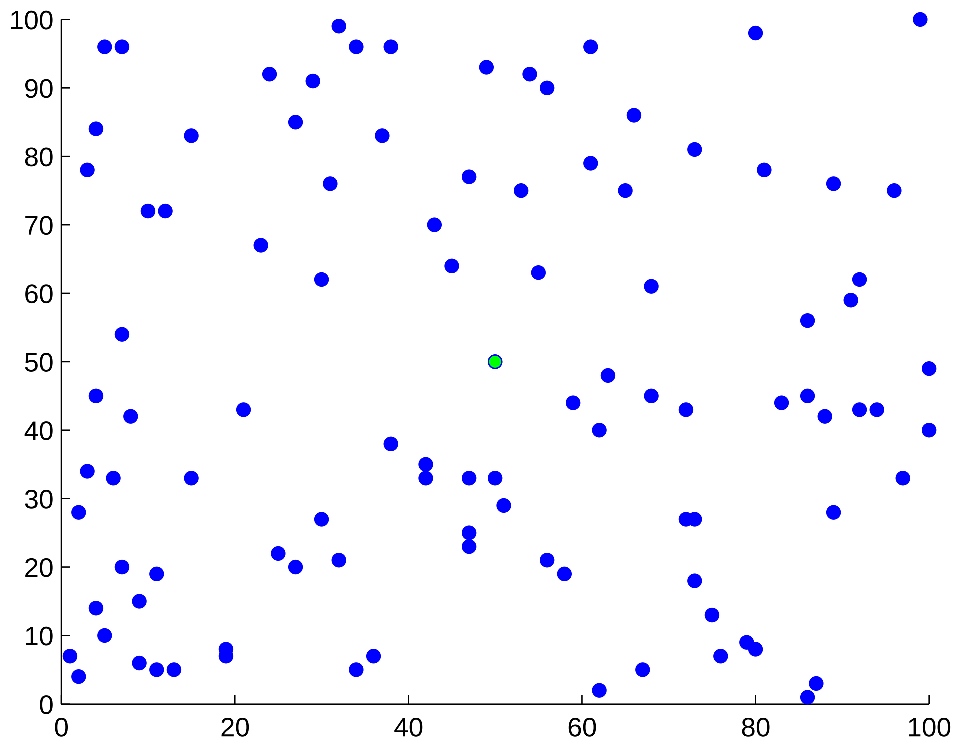
Srikanth P L received his Master's degree from the Department Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. His research interest is in the area of Web Technology, Semantic Web and Cloud Computing.

S H Manjula is currently the Chairman, Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. She obtained her Bachelor of Engineering and Masters Degree in Computer Science and Engineering from University Visvesvaraya College of Engineering. She was awarded Ph.D in Computer Science from Dr. MGR University, Chennai. Her research interests are in the field of Wireless Sensor Networks and Data mining.

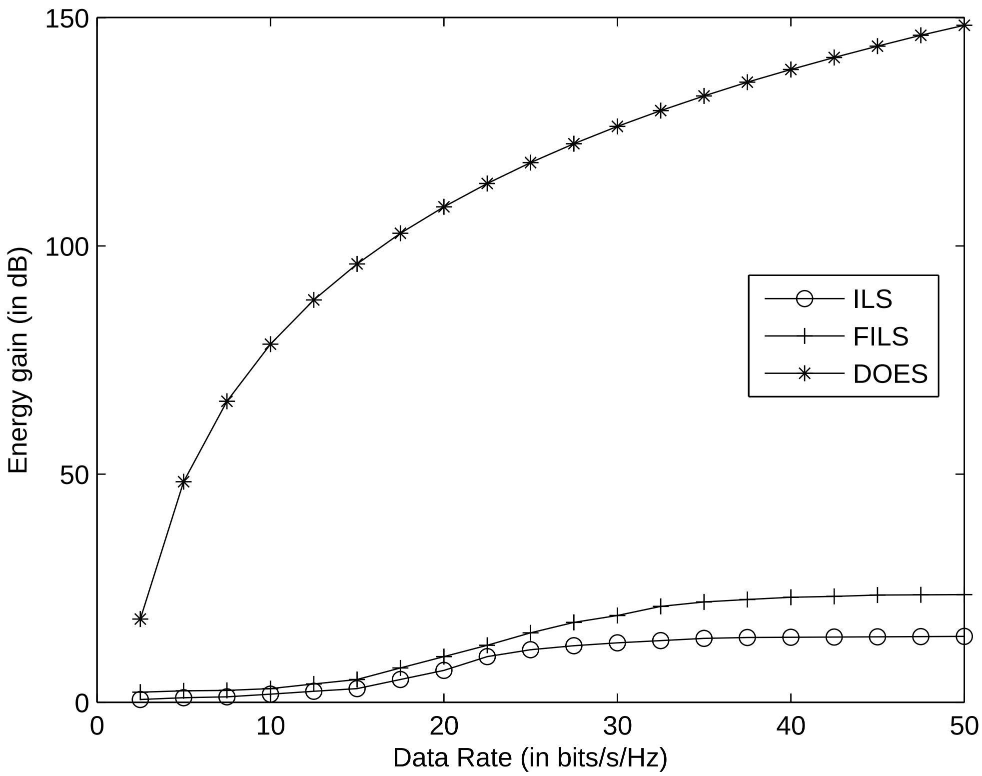
K R Venugopal is currently the Principal, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. He obtained his Bachelor of Engineering from University Visvesvaraya College of Engineering. He received his Masters degree in Computer Science and Automation from Indian Institute of Science Bangalore. He was awarded Ph.D in Economics from Bangalore University and Ph.D in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science and Journalism. He has authored 31 books on Computer Science and Economics, which include Petrodollar and the World Economy, C Aptitude, Mastering C, Microprocessor Programming, Mastering C++ and Digital Circuits and Systems *etc.*. During his three decades of service at UVCE he has over 250 research papers to his credit. His research interests include Computer Networks, Wireless Sensor Networks, Parallel and Distributed Systems, Digital Signal Processing and Data Mining.

L M Patnaik , Honorary Professor, IISc, Bangalore, is a former Vice Chancellor, Defense Institute of Advanced Technology, Pune, India. He was a Professor since 1986 with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore. During the past 35 years of his service at the Institute he has over 500 research publications in refereed International Journals and Conference Proceedings. He is a Fellow of all the four leading Science and Engineering Academies in India; Fellow of the IEEE and the Academy of Science for the Developing World. He has received twenty national and international awards; notable among them is the IEEE Technical Achievement Award for his significant contributions to High Performance Computing and Soft Computing. His areas of research interest have been Parallel and Distributed Computing, Mobile Computing, CAD for VLSI circuits, Soft Computing and Computational Neuroscience.

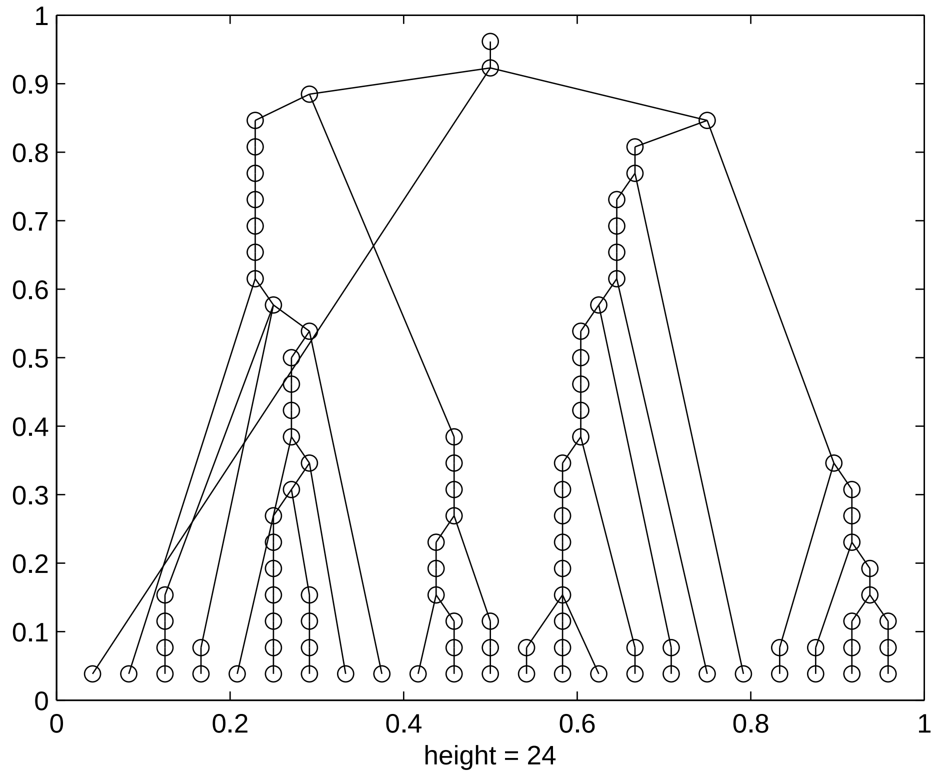

N=10
N=100
Energy Gain (in dB)
bitrate (in bits/s/Hz)
30
25
20
15
10
5
0
−5
−10
0
5
10
15
20
25
30
35
40

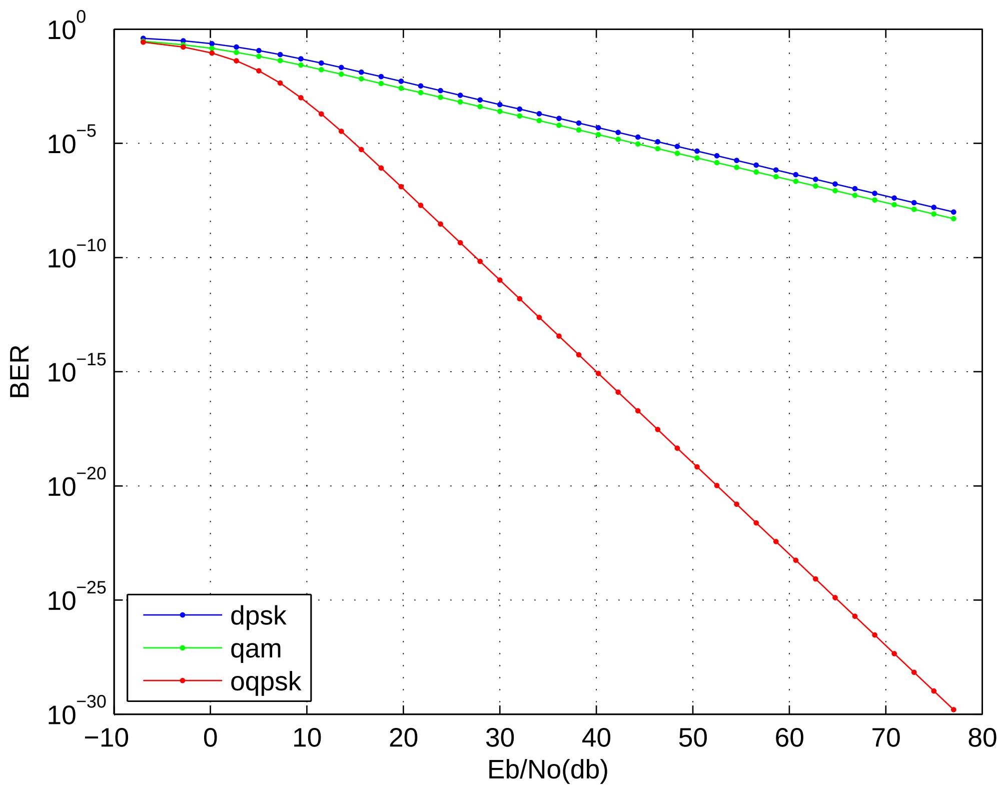


100
90
80
70
60
50
40
30
20
10
0
0
20
40
60
80
100

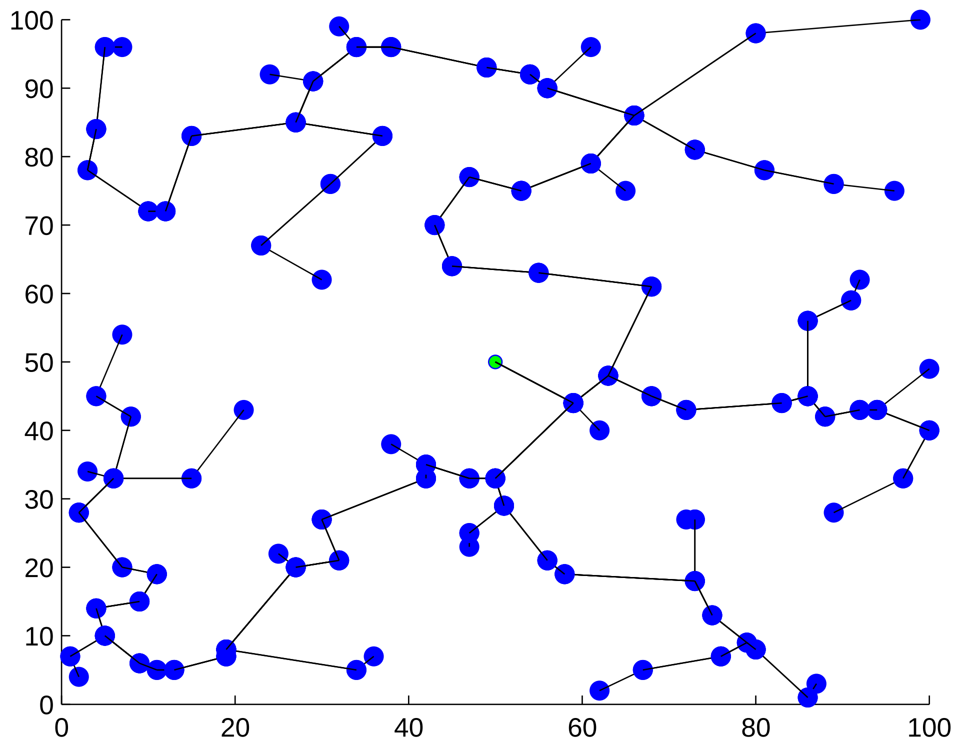


150
100
50
0
0
10
20
30
40
50
Energy gain (in dB)
Data Rate (in bits/s/Hz)
ILS
FILS
DOES

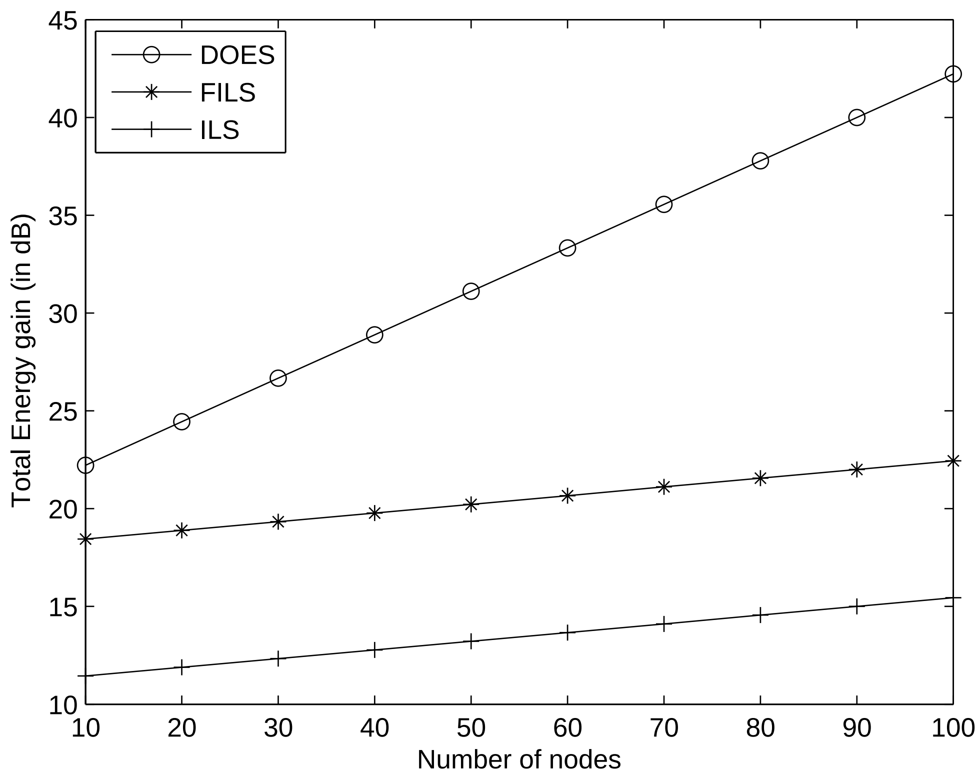

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1
height = 24



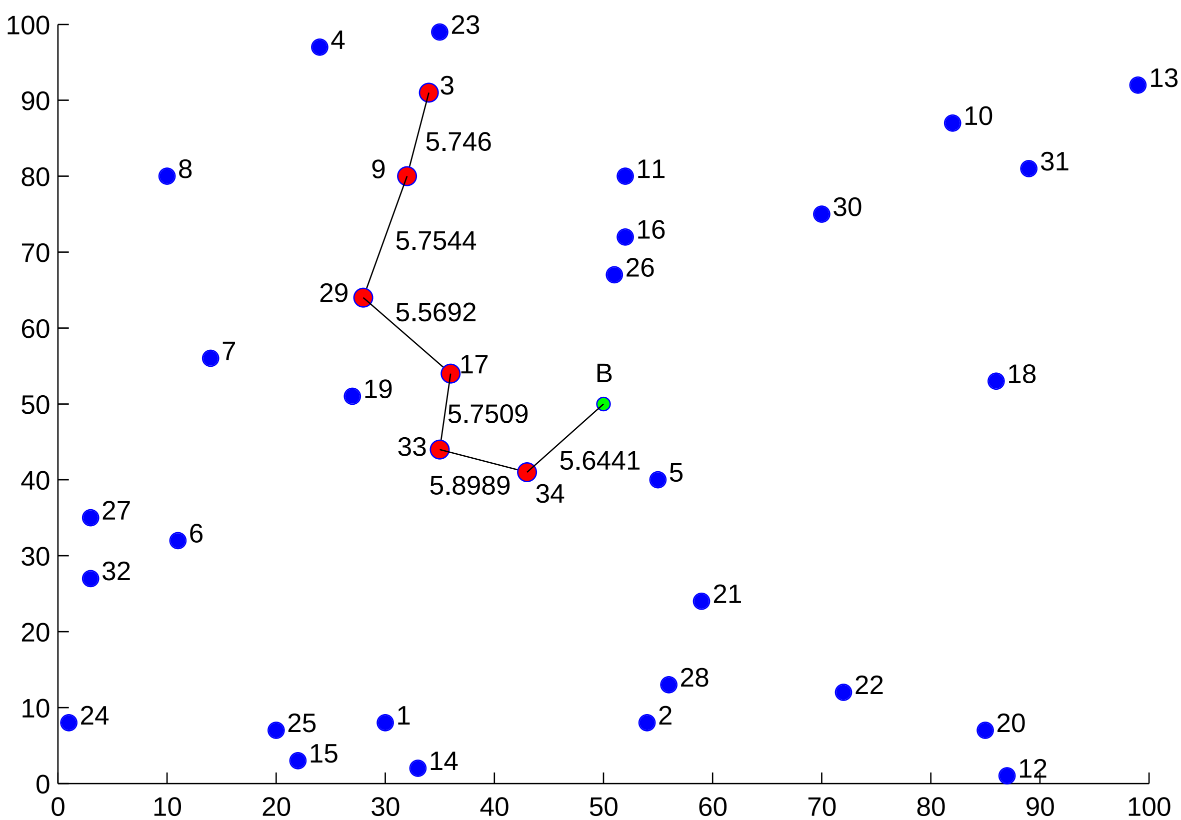
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{-20}
10^{-25}
10^{-30}
BER
−10
0
10
20
30
40
50
60
70
80
Eb/No(db)
dpsk
qam
oqpsk



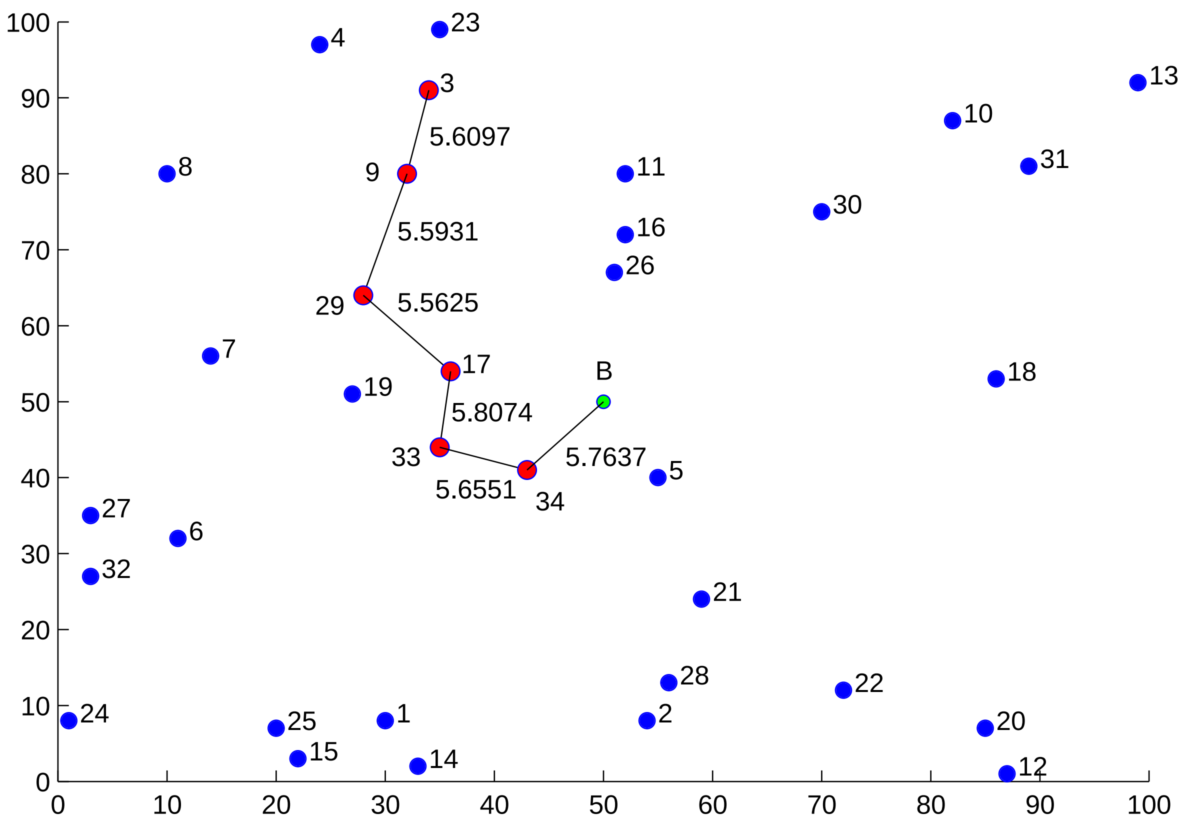
100
90
80
70
60
50
40
30
20
10
0
0
20
40
60
80
100



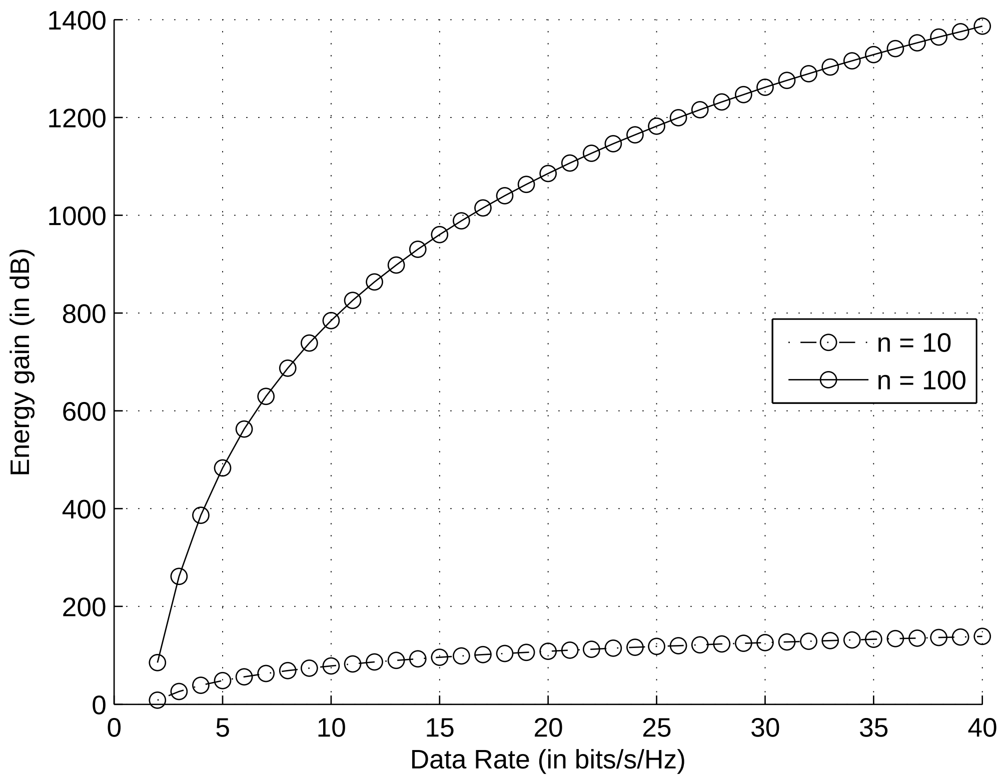
DOES
FILS
ILS
Total Energy gain (in dB)
Number of nodes
10
15
20
25
30
35
40
45
10
20
30
40
50
60
70
80
90
100



100
90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80
90
100
4
23
3
13
10
5.746
8
9
11
31
30
16
5.7544
26
29
5.5692
7
17
B
18
19
5.7509
33
5.6441
5
5.8989
34
27
6
32
21
28
22
24
25
1
2
20
15
14
12



100
90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80
90
100
4
23
3
13
10
5.6097
8
9
11
31
30
5.5931
16
26
29
5.5625
7
17
B
18
19
5.8074
33
5.7637
5
5.6551
34
27
6
32
21
28
22
24
25
1
2
20
15
14
12



1400
1200
1000
800
600
400
200
0
0
5
10
15
20
25
30
35
40
Energy gain (in dB)
Data Rate (in bits/s/Hz)
n = 10
n = 100