Journal of International Technology and Information Management

Volume 13
Issue 1 *Double Issue 1/2*

Article 4

2004

A Genetic Algorithm Assisted Hybrid Approach to Web Information Integration

Jia-Lang Seng
National Chengchi University, Taiwan

Ming-Hsiung Ying
Chin Min Institute of Technology, Taiwan

A Genetic Algorithm Assisted Hybrid Approach to Web Information Integration

Cover Page Footnote
Heterogeneity and interoperability of Web data sources represent the current key issue in Web information extraction and integration. Warehouse approach and virtual approach are the common approaches adopted to integrate heterogeneous Web data sources. However, few analytic model and cost model were developed to measure and assess the efficiency and effectiveness of either approach or a combination. Hence, a contingency model cannot be produced to assist the search engine to select and mix the warehouse method and the virtual method. In this study, we present a genetic algorithm assisted hybrid approach to aid the search engine to evaluate the cost and performance factors. We apply genetic algorithm technique to formulate a cost optimization model and compute and compare the cost of extraction and integration. The cost model is based on a collection and compilation of the property data of the query analysis and path expression of the involved Web data sources. Six property analyses are conducted and six evolution steps are created to formulate the genetic algorithm of optimization. Further, we conduct a preliminary experiment using 15 local and global Web bookstores to install and test the method. Our experimental results show that the cost optimization can be achieved with the genetic algorithm and factor analysis.

A Genetic Algorithm Assisted Hybrid Approach to Web Information Integration

Jia-Lang Seng
National Chengchi University, Taiwan

Ming-Hsiung Ying
Chin Min Institute of Technology, Taiwan

ABSTRACT

Heterogeneity and interoperability of Web data sources represent the current key issue in Web information extraction and integration. Warehouse approach and virtual approach are the common approaches adopted to integrate heterogeneous Web data sources. However, few analytic model and cost model were developed to measure and assess the efficiency and effectiveness of either approach or a combination. Hence, a contingency model cannot be produced to assist the search engine to select and mix the warehouse method and the virtual method. In this study, we present a genetic algorithm assisted hybrid approach to aid the search engine to evaluate the cost and performance factors. We apply genetic algorithm technique to formulate a cost optimization model and compute and compare the cost of extraction and integration. The cost model is based on a collection and compilation of the property data of the query analysis and path expression of the involved Web data sources. Six property analyses are conducted and six evolution steps are created to formulate the genetic algorithm of optimization. Further, we conduct a preliminary experiment using 15 local and global Web bookstores to install and test the method. Our experimental results show that the cost optimization can be achieved with the genetic algorithm and factor analysis.

INTRODUCTION

World Wide Web (WWW or Web) technology significantly changes the business and personal information computing environment. It alters the way the business and individuals search and secure data. It changes the way the data is accessed and presented. With the wireless technology emerging, Web information extraction and integration over the mobile computing platform gives the compelling reason to tackle the heterogeneity and interoperability issue with quantitative measure of cost and performance. Further, business and individuals spend ever more time and efforts searching and surfing the Web network. People place ever more attention and resource on the Web media to communicate and disseminate data and information. Heterogeneity and interoperability, as in the distributed computing, becomes one of the key issues in Web information extraction and integration.

In this paper, we present a genetic algorithm assisted method to tackle the research issue. We develop a genetic algorithm assisted optimal cost model and contingency model. We formulate a cost optimization with factor analysis to suggest a quantitative measure to select or combine the warehouse approach and the virtual approach. Six property analyses and six evolution steps are generated. Continuous query mechanism is applied to segment query request and assemble query response. We conduct a preliminary experiment using a standard set of local and global Internet bookstores to install and test the method. Experimental results show that the optimal cost can be achieved on a dynamic process. The exclusive use and hybrid use of either approach can be determined. This paper is organized into five sections. Section one introduces this research. Section two surveys the warehouse approach and the virtual approach. Section three reviews the genetic algorithm. Section four presents the cost optimization model and factor analysis. Section five describes the preliminary experiments. Section six concludes this paper with a discussion and a brief summary.

WEB INFORMATION EXTRACTION AND INTEGRATION

Web information extraction and integration is to process the Web query and operation from single or multiple Web data sources. Various researches including (Etzioni et al., 1994; Woelk et al., 1995; Arens et al., 1996; Levy et al., 1996; Garcia-Molina et al., 1997; Duschka et al., 1997; Friedman et al., 1997; Ambite et al., 1998; Beeri

et al., 1998; Cohen, 1998) have proposed different resolutions to tackle the heterogeneity and interoperability of Web sources. Most of the resolutions centers on the techniques of wrapping and mediation. Wrappers, mediators, and containers are the proposed solution. However, these techniques are based on a set of assumptions that we have (1) an ever evolving Web, (2) few poor meta-data available, and (3) the ever greater autonomy of Web sources. Hence, prior researches have created the warehouse approach and the virtual approach to work with the wrapper and mediator. These assumptions have been challenged that more and more coherent and compatible Web standards have been proposed and accepted such as XML, SOAP, and Rosetta Net. The warehouse approach and the virtual approach do not have to be mutually exclusive. The reason that the current practice is an exclusive application of independent approaches is due to an assumption that a uniform access to data obtainable from different sources available through the Web. Hence, either all necessary data is collected in a central repository before a user query is issued or the data is collected from the integrated sources dynamically during query evaluation (Vdovajk et al., 2001) (Voida et al., 2001). In this study, there can be a contingency strategy to select either independent approach or hybrid approach. The continuous query mechanism to segment and assemble query group can be applied to work with and enable the hybrid approach. Either way a cost and performance analysis model is needed to justify the decision.

Information Extraction

WWW documents are saved in the HTML/XML format. These are the semi-structured and heterogeneous data documents stored at different Web sites. Web information extraction means a process of three technical tasks to be conducted to access and retrieve the desired information including the data extraction, the data analysis, and the data presentation.

1. Data extraction: Information extraction must first extract data and deal with the problems of the data source location, the data variation, and the un-structured relationship of data objects. It needs the user intervention to decide the data source and format.
2. Data analysis: Information extraction must store and represent the hypermedia data in tables and objects. It must process user query and depend on the metadata analysis of data sources. Current methods such as the Object-Based Logical Models (OBLM) and Record-Based Logical Models (RBLM) describe the data sources by object relationships and by data fields. However, overhead of object storage and evolution is high.
3. Data presentation: Information extraction must present the query result in different data formats simultaneously. The proper amount and stream of Web pages must be determined based on the time and cost calculated.

Further, information extraction involves the process of searching and surfing over the Web. Speed, accuracy, and response time is vital. Information extraction needs an efficient and effective search strategy. There are three basic search methods including the index robot, the agent, and the wrapper/mediator.

1. Index robot: Traditional search engine utilizes the index robot to extract Web data sources. Users first enter the Web URLs. Index robot searches and builds its Web searching list. Web pages are treated as graph structures of hypertexts and hyperlinks. It adopts the depth-firth method and the breadth-first method to traverse every node and link. WWW architecture is neither a tree nor a directed cyclic graph structure. Circles result must be dealt with.
2. Agent: With the above-mentioned index robot method, each searched Web page must be sent to the search engine and be differentiated. This process causes a large amount of data to be transferred and managed. More, users are required to (1) know the difference between these Web sources, (2) spend time connecting multiple Web sources and know whether the query is processed, (3) know the different query interfaces, and (4) re-process data when conflicts and incompatible formats occur. These cause problems and inconveniences for users. Agent is an alternative. Agent's work on the collection and management of distributed data by themselves. Each agent concentrates on one type of task. Each collaborates with one another. Tasks including mapping, matching, routing, dispatching, searching, and securing. (Okada et al. 1996) proposed an architecture commonly referred to and called CAS-IG (Cooperative Agent Society for Information Gathering). The architecture has three parts: the user agent, the manager, and the machine agent.
3. Wrapper/Mediator: Wrappers are translators. A wrapper is a procedure tailored to one single Web data

source. It translates a query request and response to a relational or object format. Wrappers convert the query into local source-specific format. It can be a schema builder to compile the outside data sources into one single global schema. Mediators collaborate wrappers. A mediator examines and decides which wrapper works on which query or Web source. It dispatches works, combines results, and directs response to users.

Information Integration

As described above, two main approaches, the warehouse approach and the virtual approach, are used to integrate the heterogeneous Web data sources. The common practice is to install one approach over another, that is, an exclusive use of one approach on one Web site. Below is a brief introduction to both approaches and the reason why the current practice is an exclusive application. We add an analysis of the conditions fit to each approach.

1. Warehouse Approach: Data from multiple Web data sources is loaded into a local Web warehouse. Queries are applied with the warehoused data. Adequate performance is assured with the query response time. The cost includes the time to download all multiple Web sources and to maintain them on the local site. However, it cannot assure the extracted warehouse data is current and consistent.
2. Virtual Approach: Web data sources are not warehoused. Web queries are processed through the real time data mediation and wrapping. Data is guaranteed to be current and consistent. The cost includes the time to query, translate, and integrate these heterogeneous data sources at the run time. The disadvantage is that if the connection is broken, real time alternative is hard to generate.

Table 1 provides a set of suggested conditions when each approach should be used. We explain each condition and its justification following the presentation. These suggested conditions represent the core of the property analysis we develop. There six factors consisting of principle elements studied and discussed in standard heterogeneous information integration (HII) literature where query optimization and cost computation is presented (Tzitzikas et al., 2002) (Voida et al., 2001) (Vdovajk et al., 2001) (Vrajitoru 1998).

Table 1. Suggested Conditions of Warehouse Approach and Virtual Approach

AdoptiveApproach / Source Feature	Warehouse Approach	Virtual Approach
Average of retrieval and response time	Long	Fast
Average of update frequency	Low	High
Average of retrieval failure frequency	High	Low
Average of data structure homogeneity	Low	High
Average of retrieval contribution degree	High	Low
Average of information integration complexity	High	Low

1. The average of retrieval and response time: The average of retrieval and response time means the average time to connect to a specific Web data source and to secure the query data. If the average time is long, the portion of the Web data source should be warehoused, otherwise, the virtual approach is suggested.
2. The average of update frequency: The average of update frequency means the average of change frequency for a standard period of time. When the query requires the real time data accuracy, Web data source should be in the virtual approach. If the query inquires less frequently updated data such as the published dissertation, the warehouse approach should be used.

3. The average of retrieval failure frequency: The average of retrieval failure frequency means the average of the times that a Web data source cannot be connected over a standard period of time. If the average is higher, the warehouse approach should be used, otherwise the virtual approach is suggested.
4. The average of data structure homogeneity: The consistent degree of data scope means the degree the homogeneity of the contents of one particular Web date sources. For instance, if we query the price of computer books, the Web site queried has both the book information on computers and other un-related information such as the travel, politics, law, and economics book information. We say this Web data source has a low consistent degree and suggest a virtual approach, otherwise we suggest a warehouse approach.
5. The average of retrieval contribution degree: The average of retrieval contribution degree means that the degree that one particular Web data source can fulfill most of the query request. If the degree is low, a warehouse approach is a better choice because multiple Web data sources are un-avoidable. Poor and broken Web connections are what we try to avoid. On the other hand, if the degree is high, we recommend the virtual approach.
6. The average of information integration complexity: The complicated degree means the degree of the heterogeneity of one Web data source in terms of its Web structure, source semantics, and retrieval paths. If the source is less complicated, less time is required to integrate different data sources. The virtual approach is fine. Data currency is increased. If the source is more complicated, the warehouse approach is better than the virtual approach.

TECHNIQUE OF GENETIC ALGORITHM

Genetic algorithm (GA) is a set of probabilistic search techniques, which mimic the process of evolution. The fundamental principles of genetics lead to the development of GA. A genetic algorithm consists of five components, as described in Davis's (Davis, 1987), are:

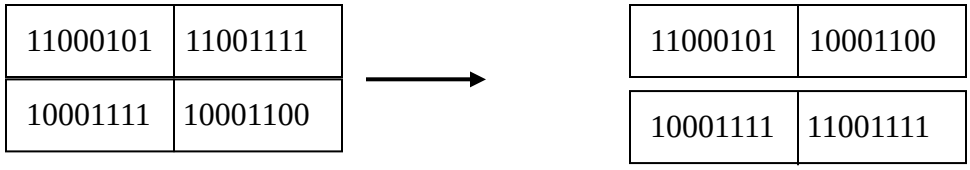
1. A method for encoding potential solutions into chromosomes.
2. A means of creating the initial population.
3. An evaluation function that can evaluate the fitness of chromosomes.
4. Genetic operators that can create the next generation population.
5. A way to set up control parameters, e.g., the population size, the probability of applying a genetic operator, and so on.

In fact, there are two operations in the traditional genetic algorithm that we adopt in this research:

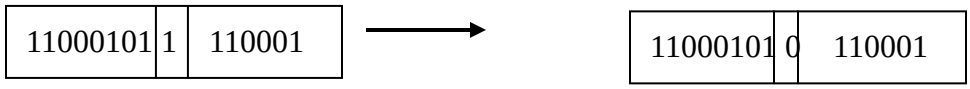
1. Genetic operations: crossover, mutation, and inversion.
2. Evolution operation: selection.

John Holland was inspired by the theory of evolution to create a computer algorithm (Holland, 1992). First, Holland used a list of zeros and ones - a bit string to encode chromosome. Computers can represent everything they do with bit string. In Holland's genetic algorithm, all evaluation functions returned positive numbers. The higher the number is, the better the chromosome is. Holland used a technique, now called the roulette wheel selection, to determine which population members are chosen for reproduction event. As shown in Figure 1, crossover operator swaps the left potions of two chromosomes. Mutation operator replaces bits on a chromosome with randomly generated bits.

Figure 1. Genetic Operators of Holland's Genetic Algorithm



(a) Crossover operation



(b) Mutation operation

The following brief outline of GA illustrates the functioning of GA, where the notation *S(t)* is the population in the i^{th} generation; $s_i(t)$ is the i^{th} member in *S(t)*; $f(s_i(t))$ is the fitness value of $s_i(t)$, and TOTFIT(*t*) is the sum of $f(s_i(t))$ for all $s_i(t)$ and *S(t)* that we will adopt and adapt in our new method.

Step 1

Generate the initial population, *S(t)*, where *t=0*. *Determine the size of the population*, POPSIZE, and the number of generations, GENER.

Step 2

Calculate the fitness value of each member, $f(s_i(t))$, for population, *S(t)*.

Step 3

Calculate the selection probability for each $s_i(t)$, where the selection probability is defined as

$P(s_i(t))= f(s_i(t))$/TOTFIT , TOTFIT = $\sum_{i=1}^{POPSIZE} f(s_i(t))$.

In our case, we have revised the formula, the $P(s_i(t))= [f(s_i(t))-Min(f(s_i(0)))]$/TOTFIT , and TOTFIT = $\sum_{i=1}^{POPSIZE} [f(s_i(t))-Min(f(s_i(0)))]$

The purpose that revises the formula is to make the higher fitness value have a higher probability of being selected with more significance.

Step 4

Select a pair of members (parents) that will be used for reproduction via the selection probability.

Step 5

Apply genetic operators (crossover, mutation, inversion) to the parents. Replace the parents with the resulting offspring to form a new population, *S(t+1)*, for generation *t*+1. If the size of the new population is equal to the POPSIZE, then go to step 6, else go to step 4.

Step 6

If current generation, *t* + 1, is equal to GENER, than stop, else go to step 2.

A GENETIC ALGORITHM ASSISTED HYBRID METHOD

In this section, we present a genetic algorithm assisted hybrid method to formulate a cost optimization model with property analysis to assess and measure the time and cost of the warehouse approach or the virtual approach or the hybrid approach. The cost optimization model works with the search engine and the continuous query mechanism to evaluate and determine if an effective information extraction and integration strategy is chosen.

Architecture

Figure 2 shows an experimental architecture of the genetic algorithm assisted hybrid method. Both the warehouse approach and the virtual approach are built. Wrapper and mediator techniques are used. An evaluation and adjustment mechanism is applied to adjust the integration approach. In specific, we first classify Web data sources into the "virtual group" and the "warehouse group". We formulate continuous queries. We compile and compute the property data of Web data sources and their status. We apply and execute the genetic algorithm through the property analyses and the evolution steps. We generate and produce the cost optimization table of results. The selection is made based on the results and contingent suggestions. A two-phase data presentation is developed to pipeline and assemble query results from the warehouse approach and the virtual approach.

The implementation of the cost optimization model implies an optimal degree of the query satisfaction reached under the user-given constraints. When the warehouse approach is chosen, it implies: (1) the Web data sources have the average of retrieval and response time longer, (2) the Web data sources have the average of update frequency lower, (3) the Web data sources have the average of retrieval failure frequency higher, (4) the Web data sources have the consistent degree of data scope lower, (5) the Web data sources have the average of retrieval contribution degree higher, and (6) the Web data sources have the complicated degree of information integration higher. When the virtual approach, it implies: (1) the Web data sources have the average of retrieval and response time faster, (2) the Web data sources have the average of update frequency higher, (3) the Web data sources have the average of retrieval failure frequency lower, (4) the Web data sources have the consistent degree of data scope higher, (5) the Web data sources have the average of retrieval contribution degree lower, and (6) the Web data sources have the complicated degree of information integration lower.

Property Analysis

Property analysis analyzes the Web source factors in information integration. As described in literature review, we generate a generic property set including (1) the average of retrieval and response time, (2) the average of update frequency, (3) the average of retrieval failure frequency, (4) the average of data structure homogeneity, (5) the average of retrieval contribution degree, and (6) the average of information integration complexity. Table 2 is an example of our preliminary experiment that we execute a property analysis. We analyze each Web data source property using the discrete distribution of weights. We derive the fitness values of each property using the translation formulas.

Figure 2. Experimental Architecture of Hybrid Approach

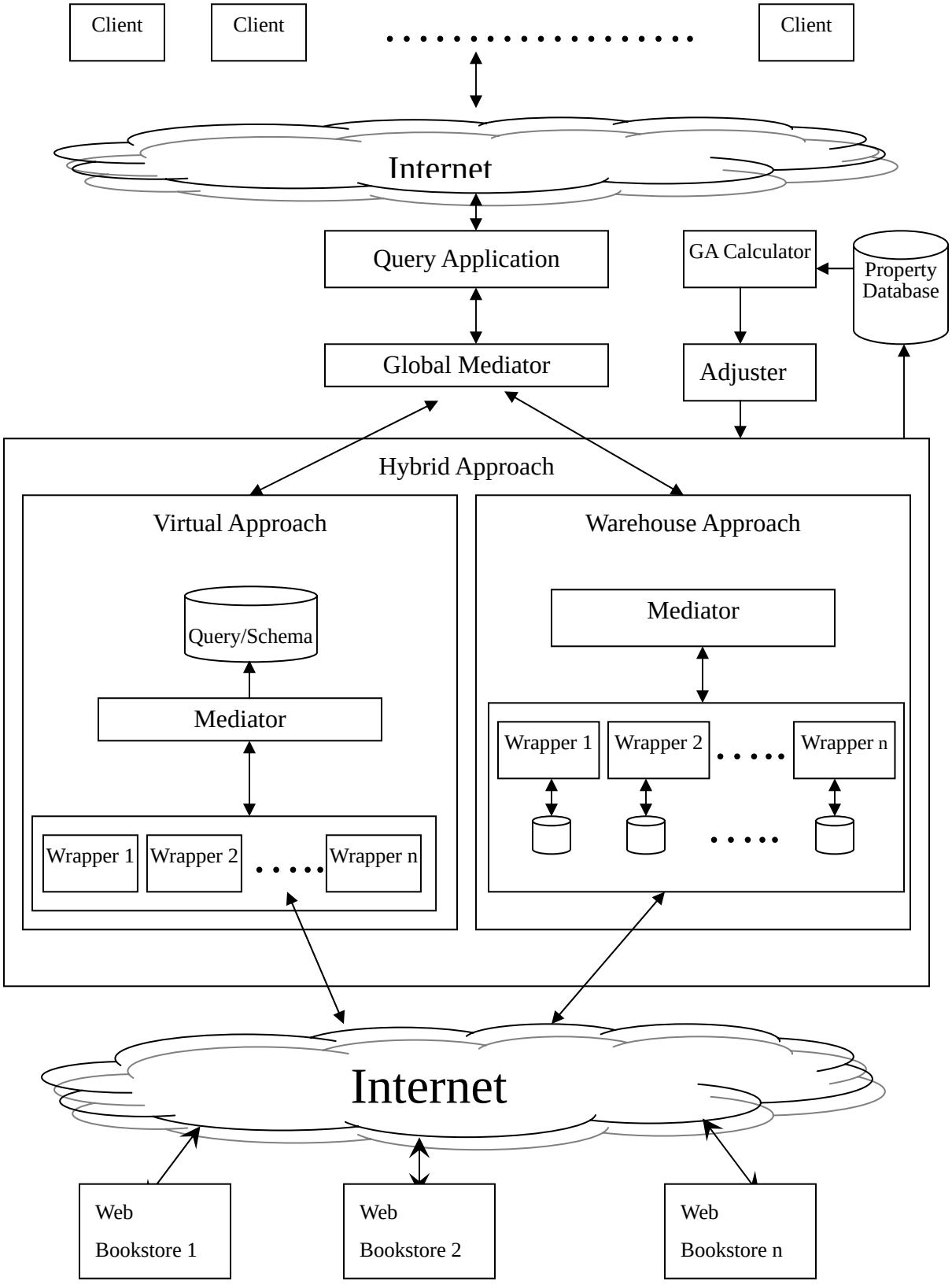


Table 2. Property Data for Each Web Data Source

		BS_1	BS_2	BS_3	BS_4	BS_5	BS_6	BS_7	BS_8	BS_9	BS_{10}	BS_{11}	BS_{12}	BS_{13}	BS_{14}	BS_{15}
C_1	V_{i1}	7.33	47.67	13	3.67	2	12.33	6.67	1.33	46.33	8	27.67	1	30	5	249
	W_{i1}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_2	V_{i2}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	W_{i2}	1	1	15	15	25	20	30	30	30	30	30	30	30	30	1
C_3	V_{i3}	0	0.14	0	0.003	0.001	0.003	0	0	0.021	0	0.012	0	0.002	0.01	0.2
	W_{i3}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C_4	V_{i4}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	W_{i4}	0.15	0.2	1	1	1	1	1	1	1	1	1	1	1	1	1
C_5	V_{i5}	0.669	0.460	0.451	0.089	0.058	0.326	0.048	0.05	0.36	0.019	0.573	0.031	0.089	0.074	1
	W_{i5}	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667
C_6	V_{i6}	1	2	9	1	8	8	1	1	10	5	8	10	10	1	1
	W_{i6}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: *gray background color is the default value, white color is the source data value

BS_1 -BS_{15}: notate the 15 Internet bookstores, i = 1…. 15

C_1~C_6: notate the six evaluation properties

C_1 : average of retrieval and response time

C_2 : average of update frequency

C_3 : average of retrieval failure frequency

C_4 : consistent degree of data scope

C_5 : average of retrieval contribution degree

C_6 : complicated degree of information integration

W_{ij} : parameter of adopting warehouse approach

V_{ij} : parameter of adopting virtual approach

Table 3. Weight of Each Property

	C_1	C_2	C_3	C_4	C_5	C_6
WTi	2	2	1	1	1	2

Each property has a different unit of measurement. We convert the property data value to the standard unit. Standard unit notates the satisfaction degree of a Web query. There are six translation formulas used to undertake these property analyses and generate the fitness values. We present each of the formulas with an explanation as follows.

Property 1(C_1):

If V_{i1}>90 sec, then V_{i1}=90

W_{i1} =(90- W_{i1})*100, V_{i1} =(90- V_{i1})*100

Property 2(C_2):
If V_{i2}>30 day, then V_{i1}=30
W_{i2} =(30- W_{i2})/30*100, V_{i2} =(30- V_{i2})/30*100

Property 3(C_3): $W_{i3} = (\sqrt{1- W_{i3}}) * 100$, $V_{i3} =(\sqrt{1- V_{i3}}) * 100$

Property 4(C_4): $W_{i4} = W_{i4}*100$, $V_{i4} = V_{i4}*100$

Property 5(C_5): $W_{i5} = W_{i5}*100$, $V_{i5} = V_{i5}*100$

Property 6(C_6): $W_{i6} =(10- W_{i6})/10*100$, $V_{i6} =(10- V_{i6})/10*100$

Property 1 notates the average of retrieval and response time and must be no greater than a user-defined level of 90 seconds. If the average of retrieval and response time has exceeded 90 seconds, the satisfaction degree equals 0. Otherwise, when the average of retrieval and response time is smaller, and the satisfaction degree is higher.

Property 2 notates the average of update frequency once and must be no greater than per 30 days (or one month) as user requests. If the average of update frequency once has exceeded 30 seconds, the satisfaction degree equals 0. Otherwise, when the average of update frequency is smaller, and the satisfaction degree is higher.

Property 3 is the average of retrieval failure frequency. If the average of retrieval failure frequency is 100% (very high), the satisfaction degree equals 0.

Property 4 is the homogeneity degree of data structure. If the homogeneity degree is higher, the satisfaction degree is higher too.

Property 5 is the average of retrieval contribution degree. If the average of retrieval contribution degree is higher, the satisfaction degree is higher too (the range of original value is from 0 to 100%).

Property 6 is the complication degree of information integration. If the complication degree of information integration is higher, the satisfaction degree is lower (the range of original value is from 0 to 10).

These properties have to be transformed to a satisfaction degree using one of these formulas. We calculate the satisfaction degree values using the fitness function described as follows.

$$\text{Fitness Function} = \frac{\sum_{i=1}^{15}[(1-a_i)*\sum_{j=1}^{6} V_{ij} * WT_j + a_i * \sum_{j=1}^{6} W_{ij} * WT_j] * 100}{}, \quad a_i = 0 \text{ or } 1$$

The a_i =0 means i^{th} Web data source that can be classified into the virtual approach group. The a_i=1 means i^{th} Web data source that can be classified into the warehouse approach group. There are 15 bits to compose one chromosome. Each bit has to be encoded by 0 or 1 and to indicate one Web data source.

In our example experiment, each data source is notated by one bit and each bit is 0 or 1 to indicate whether it is the virtual group or the warehouse group. The experiment consists of 15 Web data sources. We have 2^{15}=1024*32=32,768 combinations of variety. Though it is a large simulation, we can easily solve the problem using the genetic algorithm technique.

If a chromosome structures {a_1,a_2,….., a_{15}} has been written as {101001011000101}, it means that the 1^{th}, 3^{th}, 6^{th}, 8^{th}, 9^{th}, 13^{th}, 15^{th} Web data source can be classified into the warehouse group, and use the warehouse approach to integrate information; whereas, the 2^{th}, 4^{th}, 5^{th}, 7^{th}, 10^{th}, 11^{th}, 12^{th}, 14^{th} Web source can be classified into the virtual group, and use the virtual approach to integrate information. Each chromosome gets the fitness value by mapping every bit to a particular source property that is calculated.

IMPLEMENTATION OF GENETIC ALGORITHM

We need to implement the developed set of genetic algorithm into a process of evolution steps as discussed in literature review. The implementation is described as follows.

Step 1
Generate the initial population *S(t)* in Table 4, where *t=0.* GA parameter is GA (30, 0.975, 0.015, 1.0, 1, E), the number of generations, GENER=300.

Table 4. Initial GA Population

I	$s_i(t)$	$f(s_i(t))$	$P(s_i(t))$	Accumulative probability
001	001100100011000	74.30109	0.03804	0.03804
002	000110010001011	73.63920	0.02839	0.06642
003	100110110001111	72.05098	0.00522	0.07165
004	101111010101011	72.94028	0.01819	0.08984
005	011100100011100	74.93188	0.04724	0.13708
006	110100110000011	71.69271	0.00000	0.13708
007	011101100010111	74.81274	0.04550	0.18257
008	011001100010100	75.64617	0.05765	0.24023
009	000100010110011	72.82082	0.01645	0.25668
010	101010100111010	72.15601	0.00676	0.26343
011	100001010111011	72.76434	0.01563	0.27906
012	100011100000101	75.28389	0.05237	0.33143
013	101100100110001	73.79508	0.03066	0.36209
014	011110100011000	74.44418	0.04012	0.40221
015	111100011110100	73.95589	0.03300	0.43521
016	111000001110110	74.51656	0.04118	0.47639
017	000111010000101	75.53661	0.05605	0.53245
018	100101000000011	74.22493	0.03693	0.56938
019	101001111011111	73.68130	0.02900	0.59837
020	100100101101011	72.82786	0.01655	0.61493
021	110101011000001	74.91785	0.04703	0.66196
022	000001010000100	75.12695	0.05008	0.71204
023	011000100000110	74.39539	0.03941	0.75145
024	110011110011101	74.00224	0.03368	0.78513
025	101101001110010	73.54758	0.02705	0.81218
026	001101001111100	76.17761	0.06540	0.87758
027	101111010111010	71.76176	0.00101	0.87859
028	100011010001111	74.01786	0.03391	0.91250

029	011000101110010	73.68525	0.02906	0.94155
030	111001010010101	75.70074	0.05845	1.00000

Step 2
Calculate the fitness value of each member, $f(s_i(t))$. The fitness value of each number (see Table 4), e.g. the twenty-sixth chromosome has the optimal fitness value, and the satisfaction degree for query is 76.17761%.

Step 3
Calculate the selection probability for each $s_i(t)$, in the thesis. The selection probability is defined as $P(s_i(t))= [f(s_i(t))-Min(f(s_i(0)))]$/TOTFIT , and

$$\text{TOTFIT} = \sum_{i=1}^{\text{POPSIZE}} [f(s_i(t))-Min(f(s_i(0)))]$$

In Table 4, the twenty-sixth chromosome has the maximal fitness value, and the chromosome has the maximal probability to be selected as a parent, and the probability value is 0.0654. The sixth chromosome has the minimal fitness value, and the selected probability is 0.

Step 4
Select a pair of members (parents) that compare the accumulation probability with a random number (ranging from 0 to 1), and reproduce the chromosome into the new population.

Step 5
Apply the genetic operators (crossover, mutation, and inversion) to the parents. Replace the parents with the resulting offspring to form a new population, $S(t+1)$, for the generation t+1. If the size of the new population is equal to 30, then go to step 6, else go to step 4. In our case, every generation including 30 chromosomes, we use the higher crossover rate of 0.975 that can generate more newly structure, and prevent dropping into the local optimal solution. Our mutation rate is 0.015. If the result falls in the optimal local area, we will consider it in the computation of the global optimal area to be a possible final number.

Step 6
If the current generation, t + 1, is equal to 300, then stop, else go to step 2.

Two-Phase Presentation

Traditional query result presentation has to wait for complete query results and show them in one single window. Our method provides a two-phases presentation of query results in increments. The first phase presents the query results from the warehouse approach and from the virtual approach. The second phase pipelines both results and feeds them into the continuous query mechanism for assembly. The increments at the first phase retrieve query data from the local warehoused data sources and from the virtual data sources. Two-phase presentation allows the flexibility and efficiency of continuous heterogamous query parsing and processing.

A PRELIMINARY EXPERIMENT

We conducted a preliminary experiment to test and execute our method. The virtual approach and the warehouse approach were programmed into a set of wrappers and mediators. A Web bookstore information integration query engine was built to run with the genetic algorithm. Adjuster was created to perform the factor analysis. In this example experiment, we asked users to issue queries on book catalog and price information. Multiple heterogeneous Web sources needed to be sought and integrated. For a standard set of Internet bookstores, we got the optimal fitness value 78.63039%. Its chromosome structure was {011001001001101}. The 1^{th}, 4^{th}, 5^{th}, 7^{th}, 8^{th}, 10^{th}, 11^{th}, 14^{th} Web data sources were classified into the virtual group. We decided to use the virtual approach to test the optimization. The 2^{th}, 3^{th}, 6^{th}, 9^{th}, 12^{th}, 13^{th}, 15^{th} Web data sources were classified into the warehouse group.

We decided to use the warehouse approach to test the optimization.

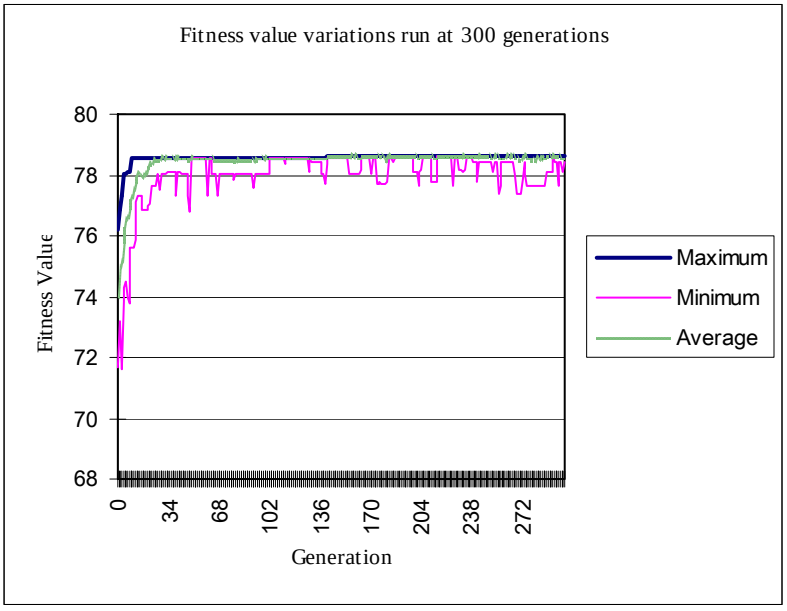
In the case where all Web bookstore data sources have chosen the warehouse approach (the chromosome structure was {111111111111111}), the fitness function value was 72.70293%. When all Web bookstore data sources have used the virtual approach (the chromosome structure was {000000000000000}), the fitness function value was 75.70533%. In this situation, we selected the combination model to extract and integrate information. The hybrid approach gave us the optimal satisfaction degree of query. Comparing with Table 2, we have found that 2th, 3th, 6th, 9th, 13th, 15th Web data sources had a longer average of retrieval and response time. We classified them into the warehouse group. But the 12th Web data source's average of retrieval and response time was faster. We classified it into the virtual group. This meant that we could not classify all Web sources at one time. We could only group them one at a time based on the property analysis. Table 5 shows the variation of the fitness function values. The fitness has changed six times. The sixth time was the final optimal solution. The 9th to 139th generations had the same fitness function value of 78.58757, and the fitness became a local optimal solution. But the 140th generation left the local optimal solution and was moved to another better optimal solution. The mutation moved it.

Table 5. Variation of Fitness Function Value

Generation	Fitness Function Value
1	76.84378
2	77.31331
3~5	78.03445
6~8	78.1096
9~139	78.58757
140~300	78.63039

Figure 3 shows the stable maximal fitness value after the 140th generation. The average fitness value rose very fast after the initial stage reaches the steady state. The minimal fitness descended gracefully with the chromosome mutation. The mutant chromosome had the minimal fitness value. Therefore, we were sure the optimal cost was found and the hybrid approach was chosen.

Figure 3. Fitness Value of Each Generation



SUMMARY

Discussion and Limitation

In this section, we discuss the implication and limitation of this research. The genetic algorithm assisted hybrid approach is innovative and incremental. However, there is the scale and scope limitation in terms of the factors compiled and applied in the property analysis in the computation and decision making. We perceive Web information integration as a process of cost optimization and property analysis. Measurement and evolution has to be conducted. The current practice is inadequate because it always chooses one approach over the other. No cost and performance is considered. No combined approach is suggested. No continuous query mechanism is applied. This study perceives the search engine in need of a cost optimization model and a contingency mechanism. Genetic algorithm is a viable approach to incorporate the heterogeneity and integration costs with generation of value evolution.

Concluding Remarks

In this paper, we have presented a genetic algorithm assisted hybrid method. The method allows the information integration to select either warehouse or virtual approach or combine both. Basic continuous query technique is used to assist the query group and assembly. Property analysis of condition and cost factors is processed in a dynamic manner. Prior researches have not been able to formulate the genetic algorithm to give the cost optimization and translation formula. The development is traceable and provable. In implementation, six evolution steps are created. We have conducted a preliminary experiment using 15 Internet bookstore Web sites to run and test the genetic algorithm assisted method. Property analyses and evolution steps are applied and measured. Experimental results show that the optimal cost was achieved and a hybrid approach was chosen. However, this is only an initial effort before any undertaking of a larger scale of experimentation and modification. In the near future, we expect to incorporate more factors of heterogeneous Web sources into the genetic algorithm method and incrementally revise the cost model. We expect to experiment with more heterogeneous Web sources to further test the robustness and generality of the method and model.

REFERENCES

Ahuja, R.K. & Orlin, J.B. & Tiwari, A. (2000). A greedy genetic algorithm for the quadratic assignment problem. *Computer Operations Research 27*(10), 917-934

Ambite, J.L. & Ashish, N. & Barish, G. & Knoblock, C.A. & Minton, S. & Modi, P.J. & Muslea, I. & Philpot, A. & Tejada, S (1998). ARIADNE: A system for constructing mediators for internet source (system demonstration). Proceedings of ACM SIGMOD Conference, Seattle, WA.

Arens, Y. & Knoblock, C.A. & Shen, W. (1996). Query reformulation for dynamic information integration. *International Journal on Intelligent and Cooperative Information Systems* (6) 2/3:99-130

Armony, M.& Klincewicz, J.G. & Luss, H. (2000). Design of stacked self-healing rings using a genetic algorithm. *Journal of HEURISTICS 6*(1) 85-105

Awad, R.M. & Chinneck, J.W. (1998). Proctor assignment at Carleton University. *INTERFACES 28*(2) 58-71

Beeri, C. & Elber, G. & Milo, T. & Sagiv, Y. & Shmueli, O. & Tishby, N. & Kogan, Y. & Konopnicki, D. & Mogilevski, P. & Slonim, N. (1998). Websuite – a tool suite for harnessing web data. Proceedings of the International Workshop on the Web and Databases, Valencia, Spain.

Carnahan, B.J. & Redfern, M.S. & Norman, B. (2000). Designing safe job rotation schedules using optimization and heuristic search. *ERGONOMICS 43*(4), 543-560

Chatterjee, S. & Carrera, C. & Lynch, L.A. (1996). Genetic algorithms and traveling salesman problems. *European*

Journal of Operations Research 93(3), 490-510

Chen, C. L. & Neppalli, R.V. & Aljaber (1996). Genetic algorithms applied to the continuous flow shop problem. *Computers and. Engineering 30*(4), 919-929

Chen, H.C. & Chung, Y.M. & Ramsey, M. (1998). A smart itsy bitsy spider for the Web. *Journal of American Society of Information Science 49*(7), 604-618

Chen, P.P. (1976). The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems 1*(1), 9-36

Chu, P.C. (1997). Beasley JE ,A genetic algorithm for the generalized assignment problem. *Computer Operations Research 24*(1), 17-23

Codd, E.F. A relation model for large shared data banks. *CACM 13*(6), 377-387

Cohen, W. (1998). Integration of heterogeneous database without common domains using queries based on textual similarity. Proceedings of ACM SIGMOD Conference, Seattle, WA

Davis, L. (1987). Genetic Algorithms and Simulated Annealing. Research Notes in Artificial Intelligence. Morgan Kaufmann, Los Altos, CA.

Dimopoulos, C. & Zalzala, A. (2000). Recent developments in evolutionary computation for manufacturing optimization: Problems, solutions, and comparisons. *IEEE Transactions on Computers 4*(2), 93-113

Duschka, O.M. & Genessereth, M.R. (1997). Query planning in informaster. Proceedings of the ACM Symposium on Applied Computing, San Joes, CA.

Ehrgott, M. & Gandibleux, X. (2000). A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum 22*(4), 425-460

Etzioni, O & Weld, D. (1994). Softbot-based interface to the internet. *CACM 37*(7), 72-76

Florescu, D. & Levy, A. & Mendelzon, A. (1998). Database Techniques for the World-Wide Web: A Survey. *ACM SIGMOD RECORD*, 59-74

Friedman, M. & Weld, D. (1997). Efficient execution of information gathering plans. In proceedings of the International Joint Conference on Artificial Intelligence, Nagoya, Japan.

Gambardella, L.M. & Taillard, E. & Dorigo, M. (1999). Ant colonies for the quadratic assignment problem. *Journal Operations Research Society 50*(2), 167-176

Garcia-Molina, H. & Papakonstantinou, Y. & Quass, D. & Rajaraman, A. & Sagiv, Y. & Ullman, J. & Widom, J. (1997). The TSIMMIS project: Integration of heterogeneous information sources.

Gupta, J. & Sexton, R.S. & Tunc, E.A. (2000). Selecting scheduling heuristics using neural networks. *INFORMS Journal Computing 12*(2), 150-162

Holland, J.H. (1992). Adaptation in Natural and Artificial System. University of Michigan Press, Ann Arbor, MI; reprint by MIT press, Cambridge, MA.

Hwang, C.P. & Alidaee, B. & Johnson, J.D. (1999). A tour construction heuristic for the traveling salesman problem. *Journal Operations Research Society 50*(8), 797-809

Kwan &. Kwan & Wren, A. (1999). Driver scheduling using genetic algorithms with embedded combinatorial traits. *Lecture Notes Economy Mathematics, 471*, 81-102

Levy, A.Y. & Rajaraman, A. & Ordille, J.J. (1996). Querying heterogeneous information sources using source descriptions. Proceedings of the International Conference On Very Large Data Bases, Bombay, India.

Lin, C.H. & Shen, S.Y. & Yeh, Y.J. (2001). Dynamic optimal control policy in advertising rice and quality. *International Journal of System Science 32*(2) , 175-184

Okada, R. & Lee, E.S. & Shiratori, N. (1996). Agent-based Approach for Information Gathering on Highly Distributed and Heterogeneous Environment. Proceedings of International Conference on Parallel and Distributed System, pp.80-87

Martin-Bautista, M.J. & Vila, M. & Larsen, H.L. (1999). A fuzzy genetic algorithm approach to an adaptive information retrieval agent. *Journal of American Society Information Science 50*(9), 760-771

Seng, Jia-Lang (2003). A simple set representation of functional and non-functional requirements. *Journal of Information Technology and Information Management*

Smith, M.P. & Smith, M. (1997). The use of genetic programming to build Boolean queries for text retrieval through relevance feedback. *Journal of Information Science 23*(6), 423-431

Sndararajan, R. & Azarm, McCluskey, P. (1999). A stress model for multiobjective design optimization of a power electronic module. *Mechanic Structure Machine 27*(2), 163-183

Tzeng, G.H. & Chen, Y.W. (1999). The optimal location of airport fire stations: A fuzzy multi-objective programming and revised genetic algorithm approach. *Transport Plan Tech 23*(1), 37-55

Tzitzikas, Y., Spyratos, N., and Constantopoulos, P. (2002). Query Translation for Mediators over Ontology-Based Information Sources. Proceedings of the Second Hellenic Conference on Artificial Intelligence (SETN2002), pp.423-436

Vdovjak, R., and Houben, G. (2001). RDF-Based Architecture for Semantic Integration of Heterogeneous Information Sources. Proceedings of the Workshop on Information Integration on the Web 2001, pp.51-57

Voida, Stephen, Elizabeth D. Mynatt, Blair MacIntyre, Gregory M. Corso (2002). Integrating Virtual and Physical Context to Support Knowledge Workers. *IEEE Pervasive Computing 1*(3), pp.73-79

Vrajitoru, D. (1998). Crossover improvement for the genetic algorithm in information retrieval. *Information Processing Management 34*(4), 405-415

Wilson, J.M. (1997). A genetic algorithm for the generalized assignment problem. *Journal of Operations Research Society 48*(8), 804-809

Wittkemper, H.G. & Steiner, M. (1996). Using neural networks to forecast the systematic risk of stocks. *European Journal of Operations Research 90* (3), 577-588

Woelk, D. & Bohrer, B. & Jacobs, N. & Ong, K. & Tomlinson, C. & Unnukrishnan. C. (1995). Carnot and infoseleuth: Database technology and the World Wide Web. Proceedings of ACM SIGMOD Conference, 443-444, San Jose, CA.