Wuhan University Journal of Natural Sciences

2009, Vol.14 No.2, 115-118

Article ID 1007-1202(2009)02-0115-04

DOI 10.1007/s11859-009-0204-y

Optimization of Web Search Engine and Its Application to Web Mining

□ CHEN Hao[1,2], ZOU Beiji[1,3†], BIAN Naizheng[2†]

1. School of Information Science and Engineering, Central South University, Changsha 410083, Hunan, China;

2. Software School, Hunan University, Changsha 410082, Hunan, China;

3. Department of Computer Science and Technology, Huaihua University, Huaihua 418008, Hunan, China

Abstract: With the explosive growth of information sources available on the World Wide Web, how to combine the results of multiple search engines has become a valuable problem. In this paper, a search strategy based on genetic simulated annealing for search engines in Web mining is proposed. According to the proposed strategy, there exists some important relationship among Web statistical studies, search engines and optimization techniques. We have proven experimentally the relevance of our approach to the presented queries by comparing the qualities of output pages with those of the original downloaded pages, as the number of iterations increases better results are obtained with reasonable execution time.

Key words: Web mining; genetic algorithm; simulated annealing

CLC number: TP 393

Received date: 2008-06-04

Foundation item: Supported by the National Natural Science Foundation of China (60673093)

Biography: CHEN Hao(1975-), male, Ph. D. candidate, Lecturer of Hunan University, research directions: Web mining, software testing, service-oriented architecture and its technology. E-mail: xschenhao@gmail.com

† To whom correspondence should be addressed. E-mail:bjzou@vip.163.com

0 Introduction

With the explosive growth of information sources available on the World Wide Web, it has become increasingly necessary for users to utilize automated tools to find the desired information resources, and to track and analyze their usage patterns. These factors induce the necessity of creating server-side and client-side intelligent systems that can effectively mine knowledge. Web mining can be broadly defined as the discovery and analysis of useful information from the World Wide Web. This describes the automatic search of information resources available online.

In recent years, genetic algorithm (GA) has been used, mainly in the search, optimization, and description of Web mining. Here, we describe some of the attempts. A GA-based search to find other relevant homepages, given some user-supplied homepages, has been implemented in G-Search[1]. Web document retrieval by genetic learning of importance factors of HTML tags has been described in Ref.[2]. In Ref.[3], Boughanem *et al* developed a query reformulation approach using GA, in which a GA generates several queries that explore different areas of the document space and determine the optimal approximate matching in documents. Automatic Web page categorization and updating can also be performed using GA[4]. It focuses on the problem of index page synthesis where an index page is a page consisting of a set of links that cover a particular topic in Refs.[5,6]. Here, GA may be used to predict user preferences, dynamic optimization, and evolution of Web pages.

The aim of this paper is to define a steady state genetic simulated annealing algorithm (GSAA) that evolves a population of pages. GA is randomized search and optimization techniques guided by the principles of evolution and natural genetics. It is an efficient, adaptive and robust search process that can produce near optimal solutions, and has a large amount of implicit parallelism. Although GA is powerful in global search, it tends to converge prematurely in the practical application, and has low search efficiency in the later evolutionary process. Simulated annealing (SA) originates from the method of the statistical physics and is first employed by Kirkpatric to solve the optimization problem. It has a more powerful local search ability than GA, but it depends more on parameters. GSAA is an optimal algorithm combining GA with SA. GA is weak in local search but powerful in global search while SA is weak in global search but powerful in local search[7-10].

1 Map the Web Search Problem to GSAA

GSAA combines the concepts described in the previous section with those of a steady state GA. An individual in the population is a Web page that can be numerically evaluated with a fitness function. Initially, the first individuals are mostly generated with a heuristic creation operator which queries standard search engines to obtain pages. Then, the individuals can be selected according to their fitness, and can generate offspring with crossover /mutation operators.

1.1 Fitness Function

The fitness function $f(*)$ that evaluates Web pages is a mathematical formulation of the user query and numerous evaluation functions. We have used function $f(*)$, similar to the evaluation functions used in standard search engines.

Definition 1 Link quality $F(L)$

$$F(L)=\sum_{i=1}^{n}\Sigma K_i$$

where n is the total number of input keywords, ΣK_i is the mean number of occurrence in link and $K_1, K_2, K_3, \cdots$ are the keywords given by the user.

Definition 2 Page quality $F(P)$

$$F(P)=\sum_{j=1}^{m}F_j(L)$$

where m is the total number of links per page.

Definition 3 Mean quality function M_{q}

$$M_{\mathrm{q}}=\frac{F_{\max}(P)+F_{\min}(P)}{2}$$

where $F_{\max}(P)$ and $F_{\min}(P)$ are the maximum and minimum values of the page qualities, respectively after applying the GSAA. It should be noted that the biggest value of $F_{\max}(P)$ is $m\times n$, and the least value of $F_{\min}(P)$ is zero. Hence, the upper limit of M_{q} is ($m\times n$)/2. Application of the GSAA to Web pages will increase the qualities of some pages and decrease those of the others.

During the reproduction phase, each individual is assigned a fitness value derived from its raw performance measure given by the objective function. This value is used in selecting fitter individuals. The individuals with better fitness values in the whole population have a high probability of being selected for mating, whereas the individuals with worse fitness values have a correspondingly low probability of being selected. The fitness function used in our GSAA is given by $f(x)=\mathrm{e}^{M_{\mathrm{q}}}$.

1.2 Selection Operator

We use "Stochastic Universal Sampling" as our selection method[11]. A form of stochastic universal sampling is implemented by obtaining a cumulative sum of the fitness values, and generating popsize equally spaced numbers between 0 and sum($f(*)$). Thus, only one random number is generated, all the others being equally spaced from that point. The index of the individuals selected is determined by comparing the generated numbers with the cumulative sum vector. The probability of an individual being selected is then given by

$$P(i)=f(i)/\sum_{i=1}^{\text{popsize}}f(i)$$

where $f(i)$ is the fitness of individual (Web page) i and $P(i)$ is the probability of that individual being selected.

1.3 Crossover Operator

The following is the definition of the crossover operator:

Step 1 Generate a random number r_{c} in the interval [0,1] that equals to the number of pages. If $r_{\mathrm{c}} < P_{\mathrm{c}}$, then conduct the following crossover operation: $V_1'=r_{\mathrm{c}}\times V_1, V_2'=r_{\mathrm{c}}\times V_2$, in which V_1, V_2 are both parent pages, and V_1', V_2' are both offspring.

Step 2 Compute the fitness value $f(V_1')$ and $f(V_2')$, if $\min\{1,\exp(-(f(V')-f(V))/T_k)\} >$ random [0,1], then accept the new page. T_k is the annealing temperature of the kth time.

1.4 Mutation Operator

A further genetic operator, mutation is applied to the new chromosomes, with a set probability P_{m}. Mutation causes the individual genetic representation to be changed according to some probabilistic rules. Mutation is generally considered to be a background operator that ensures the probability of searching a particular subspace if the problem space is bigger than zero. This has the effect of tending to inhibit the possibility of converging to a local optimum, rather than the global optimum. The following is the definition of the mutations operator.

Step 1 Generate a random number r_m in the interval [0,1], if $r_m < P_m$, then conduct the following mutation operation on the individual V_i: Use the approach to randomly generate (integer) *w* ranging from 1 to the total number of the links per page to determine the mutation point, then use w to substitute K_w in V_i.

Step 2 Determine whether the solution is accepted or not after the mutation operator according to Step 2 in the crossover operator.

1.5 Termination Condition

When the temperature T_0 in the process of cooling no longer falls, the annealing process will naturally stop.

1.6 The Solution Process of GSAA

The solution process of GSAA is as follows:

Step 1 Get the user request and initialize control parameters: $k = 0$, the initial temperature T_0, the size of the population popsize;

Step 2 Initialize parent population $P(k)$;

Step 3 Evaluate the fitness value $f(x)$ of each individual in population $P(k)$, and then select $P(k)$ to generate the parent population $F(k)$;

Step 4 Crossover $F(k)$ to generate the crossover population $C(k)$, and then mutate $C(k)$ to generate the mutation population $M(k)$;

Step 5 Generate the next population

$P(k+1) = F(k) \cup M(k)$;

Step 6 When the termination condition is satisfied, the annealing process will naturally end and give the outputs to the user; otherwise, $T_{k+1} = T_k \times (1 - \frac{k}{\text{popsize}})$, $k = k+1$, go to Step 3.

2 Experiment Results

2.1 Settings of the Experiment

The proposed GSAA has been implemented using C++ language. The obtained program was tested using PC PIV 2.8 GHz processor, 512 MB RAM, and HDD of 80 GB. For each problem, we use 300 downloaded pages from the standard four search engines (Yahoo, Google, Baidu, and MSN). They were stored in the HDD for further operations. Tabulated results are averaged over 5 runs.

2.2 Experiment Results

Referring to Fig.1 of query No.1, we note that, if the population size is too small (e.g. 20 pages), the GSAA decreases the quality of the results. If the population size is large (e.g., 100 pages), the selection operator does not concentrate the search on important pages. We can also note that when P_c is large (i.e., $P_c = 0.75$) the search algorithm spends more time with unsuccessful exchanging links. To improve the obtained results at these P_c, the number of iterations should be increased. Figs. 2-4 illustrate these phenomena. As a result, the execution time will rapidly increase as shown in Fig.4.

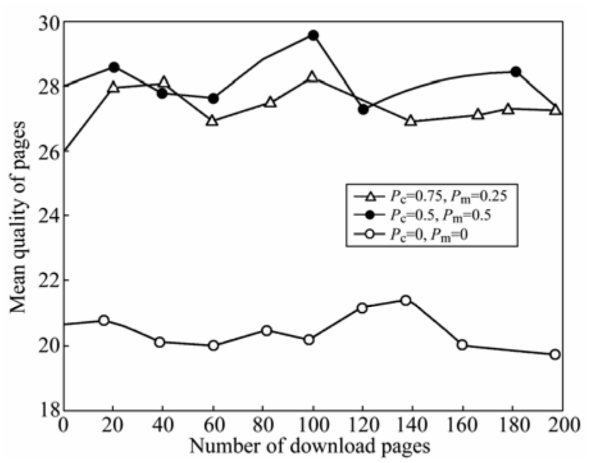

Fig.1 Population mean quality for different values of popsize at 3 000 iterations

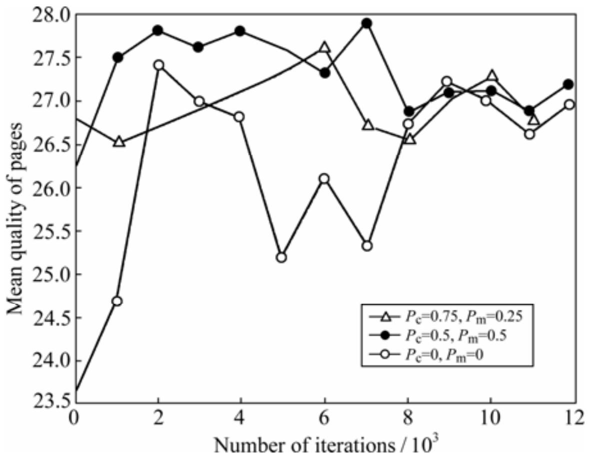

Fig.2 Population mean quality for different values of iteration times at 20 pages

3 Conclusion

In this paper, we have proven experimentally the relevance of our approach on the presented queries by comparing the qualities of output pages with those of the original downloaded pages. As the number of iterations increases better results are obtained still with reasonable

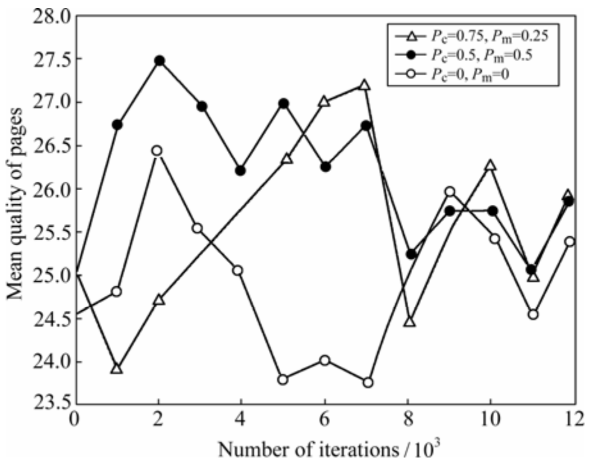

Fig.3 Population mean quality for different values of iterations at 250 pages

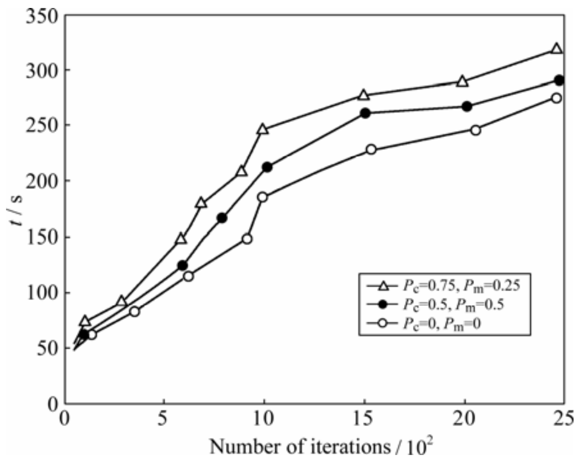

Fig.4 Variation of time with number of iterations using 250 pages

execution time. The small size of pages P_c limits the chances of improving the page qualities and reducing execution time at a specified number of iterations. It should be noted that the results depend on the preparation methods of the constituting pages under test.

References

[1] Crestani F, Pasi G. *Soft Computing in Information Retrieval*: *Techniques and Application*[M]. Heidelberg: Springer-Verlag, 2000: 154-164.

[2] Kim S, Zhang B T. Web Document Retrieval by Genetic Learning of Importance Factors for Html Tags[C]// *Int'l Workshop on Text and Web Mining*. New York: Springer-Verlag, 2000: 13-23.

[3] Boughanem M, Chrisment C, Mothe J, *et al*. Connectionist and Genetic Approaches for Information Retrieval[C]// *Soft Computing in Information Retrieval*:*Techniques and Applications*. Heidelberg: Springer-Verlag, 2000:102-121.

[4] Loia V, Luongo P. *An Evolutionary Approach to Automatic Web Page Categorization and Updating*[M]. Singapore: Springer- Verlag, 2001: 292-302.

[5] Martino V, Mililotti M. Sub-Optimal Scheduling in a Grid Using Genetic Algorithm[J]. *Parallel Computing*, 2004, **30**(5): 553-565.

[6] Etzioni O, Perkowitz M. Adaptive Web Sites:An AI Challenge[C]// *Proc of the* 15*th Int'l Joint Conf Artificial Intelligence*. San Francisco:Morgan Kaufmann Publishers, 1997: 16-21.

[7] Shu Wanneng, Zheng Shijue. A Parallel Genetic Simulated Annealing Hybrid Algorithm for Task Scheduling[J]. *Wuhan University Journal of Natural Sciences*, 2006, **12**(5): 1378-1382.

[8] Abraham A, Buyya R. Nature's Heuristics for Scheduling Jobs on Computational Grids[EB/OL]. [2007-12-21]. *http://www.softcomputing.net/adcom.pdf.*

[9] Zhang Jiangshe, Xu Zongben, Liang Yi. Global Annealing Genetic Algorithm and Its Convergence Well Necessary Condition[J]. *Science in China*, 1997, **27**(2): 154-164.

[10] Lu Shan, Chen Tong, Xu Shijie. Optimal Lambert Transfer Based on Adaptive Simulated Annealing Genetic Algorithm[J]. *Journal of Beijing University of Aeronautics and Astronautics*, 2007, **33**(10): 1191-1195(Ch).

[11] Wang Xia, Zhou Guobiao. Strong Convergence of Global Annealing Genetic Algorithm[J]. *Mathematica Applicata*, 2003, **16**(3): 1-7.

□