Using PSO Algorithm to Optimize Linking Structure of E-commerce Sites

Dingying Tan, Pingping Chen, Xiufeng Liu, Huiling Liu, Bin Sun

Guangzhou University of Chinese Medicine

Guangzhou, China e-mail: computertan@163.com

Keywords: Particle Swarm Optimization; E-commerce; Web-site structure；Linking

Abstract. In order to facilitate user access to e-commerce site, improve site access rate, meet the needs of users and maintain website performance, this paper is to establish a structural optimization model and to provide algorithm for ecommerce business by starting from structural optimization and adjustment of e-commerce sites links. The simulating e-commerce sites is used as the example to test, and the experimental results show that the algorithm more efficient operation and optimized fitness value greater than the original fitness value, particle swarm optimization (PSO) can be used to optimize the structure of e-commerce sites links, adjust the site structure automatically, therefore it can improve the performance of the website maintenance, which has certain reference value for the application.

Introduction

With the rapid development of web and network technology, e-commerce sites have become a popular platform to facilitate the communication between enterprises and their customers. Many researches have been done on how to provide an easier and a more convenient way for the trade between customers and enterprises through e-commerce sites. As to an ecommerce site with complicated structures and lots of web pages, the website design and dynamic adjustment and maintenance is very important. The structural design of e-commerce sites directly influences operational behavior of users and significantly determines if a site can attract customers and guide their consumption. If all these were done manually, it will not only make heavy workload for management personnel, but also create more difficulties to grasp the needs of customers accurately.

This paper proposed a structural optimization model and provided an algorithm to optimize and adjust the structure of website links. The model is based on visit behavior of customers, with a condition that the basic structure was not damaged. The algorithm is developed to optimize the linking structures, taking linking pages as constraint conditions and maximizing visiting-frequency as the optimal objective. The rest of this paper is organized as follow.

The Optimized Model Of E-Commerce Linking Structures

A. Structures of websites

Website is a complex network structure. There are many paths to different pages, and each path to every page will vary too. Generally speaking, the website linking will be divided into the basic linking and the additional linking [1]. Basic linking reflects the basic structure, and is linked from the top web page to the down page along the linking structure of classified commodities and is the most important linking that cannot be deleted. Additional linking is set just for the conveniences of searching, and thus is linked from the same level or to the upper page, and is adjustable.

B. The optimized model of E-commerce linking structures

E-commerce sites have lots of pages, and linking structures are very complex. The linking structure can be illustrated by digraphs with directions, using adjacency matrix to store data.

1) The linking structure: Using the data structure graph theory [2], we assume that matrix M(n) will represent website linking structure, including M for n * n order matrix, n for numbers of pages, x_{ij} as the j line elements on the i row in the matrix. When i is [0, n - 1], and x_{ij} is 0, then the linking

Published by Atlantis Press, Paris, France.

from page i to j does not exist, whereas when x_{ij} is 1,then the linking from page i to j shows the best linking status. M(n) can be expressed as:

$$x_{ij}=\begin{cases}1 & \text{page } i \text{ to } j \text{ links exist} \\ 0 & \text{page} i \text{ to } j \text{ does not exist}\end{cases} \quad i, j \in [0,n-1] \tag{1}$$

2) The visiting frequency of linking: The visiting frequency from page i to j, can be calculated as the following formula (formula 2) below:

$$f_{ij}=\frac{u_{ij}}{\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}u_{ij}} \quad i,j \in [0,n-1] \tag{2}$$

f_{ij} is the visiting frequency from page *i* to page *j* , and u_{ij} expresses visiting times from page *i* to page *j*.

3) The Vi out-degree

The hyper linking numbers from *Vi* to other pages of this site.

4) The optimized linking model

In order to optimize the linking, we will take the maximization of visiting frequency of linking as our optimal goal, which has two constraint conditions. The first is: basic linking structure remains. The basic linking structure reflects the basic classification and structural levels of commodities, thus its structure should not be changed. The second is: the numbers of a single web page linking will be limited. Too many linking in a single page will result the information overloaded and in turn will affect the sharpness of the page. The linking numbers correspond to out-degree of the nodes of a page. The out-degree of *i* page is the aggregate number of the line *i* of the matrix M (n), i.e. $d_i=\sum_{j=0}^{n-1}x_{ij}$.

The optimized linking model will be illustrated as the following formula (formula 3) below.

Optimized function:

$$Q=\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}f_{ij}x_{ij} \quad i,j \in [0,n-1] \tag{3}$$

Constraint conditions

$$x_{ij}-a_{ij}\geq 0 \quad i,j \in [0,n-1] \tag{4}$$

$$\sum_{j=0}^{n-1}x_{ij}\leq d \quad i \in [0,n-1] \tag{5}$$

Among them, x_{ij} is the linking matrix M (n), and a_{ij} as the basic linking matrix N (n), and f_{ij} as the element of visiting frequency matrix F (n), *d* for maximum out-degree of linking, *n* for numbers of site page.

When the basic linking remains, we take into account numbers of page linking, assuming the out-degree is maximized, therefore the overloaded information will be avoided and the linking quality will be improved and the foundation for website optimization will be provided.

Using Pso to Optimize the Linking

C. Thoughts *of particle swarm optimization algorithm*

The underlying principle of Particle Swarm Algorithm [3]: a group consist of m particles fly with certain speed in D dimension, and when every particle is searching its space, it has searched its best historic position and the best historic position of other particles, then it changes its position within this position framework, and then the final position of the particle is the optimization solution. Hence, the current position of the particle will influence the fitness of the target function, which determines the good or the bad degree of each particle. The flight direction and speed of a particle will be determined by a velocity vector.

Originally particle swarm algorithm was applied to solve problems of continuous space optimization. Kennedy and Eberhart proposed the discrete binary particle swarm optimization algorithm to solve the problem of space discreteness [4]. Using this algorithm, the need for adjusting parameter is less, and because it has good generality, robustness, fast convergence speed, therefore it has widely applied in all kinds of optimization problems. This article will use algorithm to solve the optimization problem of e-commerce linking.

In binary particle swarm algorithm, each value of the particle's position vector is 0 or 1, and the speed of the particle will be defined as the change probability of the position of the particle, then the replaced formula(references [4]) will be presented as follows (formula 6):

$$x_{im}^{k+1} = \begin{cases} 1 & sig(v_{im}^{k+1}) \ge p \\ 0 & others \end{cases} \qquad (6)$$

In formula 6, the value of x_{im}^{k+1} is limited to 1 or 0, and forv_{im}^{k+1}, $|v_{im}^{k+1}|<v_{max}$. The speed of particles will be used to determine its value is 0 or 1 in its corresponding position. When the speed is fast, the probability that its corresponding position is 1 is big; likewise, the probability of corresponding position to 0 is big. *p* represents a number from 0 to 1, and is a boundary number. When the probability is greater than *p,* the particle position vector is 1; whereas when it is less than *p* then the particle position vector is 0. *Sig* is sigmoid function[4], and is defined as:

$$sig(v_{im}^{k+1}) = \frac{1}{1+e^{-v_{im}^{k+1}}} \qquad (7)$$

From formula 7, we know that the greater the speed, the sigmoid function value is more close to 1; vice versa, the smaller, the sigmoid function value will be more close to zero. For v_{max} value, Kennedy reckons that when the value of v_{max} is 6, then good result can be achieved, and at this moment the Sigmoid function value will range in [0.0025, 0.9975].

D. *The application of binary particle swarm optimization algorithm to linking optimization*

1) The coding of particles: The Coding directly determines the accuracy of an algorithm. The website structure will be saved as a matrix, and the coding of particles will be turned to a matrix form.

2) The initialization of particles: Particles will be established under the basic linking constraint condition. And when one element of the basic linking is 1, then the corresponding position value of initial particles is 1; Meanwhile, when the value of the visiting frequency matrix of this position is not 0, then the value of the corresponding position will be set to 1, so that the particle will be in a near optimal solution at the initial stage. Then use the random functions 0-1 to assign to the particles elements with 0, and finally to make judgments on constraint conditions.

3) The processing of restrictions: There are three restrictions for linking optimization, and they are formula 4, 5 and the maximum speed constraint respectively. In the process of generating and replacing the initial particles, the unsuitable particles will be dealt as:

a) The basic link restriction

When the particle is not suitable for the basic linking restriction, the value of particles matrix elements which are not suitable for x_{ij}-a_{ij}<0 restriction should be set as the basic value of the matrix a_{ij} elements, so that the initial produced and replaced particles will meet the requirement of basic linking restriction.

b) The largest out-degree restriction

When the home page and the largest out-degree restrictions are used, the latter refers to the maximum out- degree of other pages except the home page. When the line *i* appears $\sum_{j=0}^{n-1} x_{ij} > d$, 0-1 integer will be generated randomly, and when the value of a random number is 0, starting from the zero element of line *i*, the value of the element x_{ij} =1 but a_{ij} is not 1 will be set to 0 until the out-degree *d* appears; when the value of a random number is 1, change the searching direction, the procedure of modifying elements is the same as the procedure when the random value is 0.

c) Maximum speed restriction

When the renewal speed matrix of v_{ij} element is faster than v_{max}, the value of the element is set to be $-v_{max}/2$, but when the element v_{ij} is less than -v_{max}, the value of the element will be set to be $v_{max}/2$, when the absolute value of the speed is less than 0.0001, and the speed is bigger than 0, the value will be set to - $v_{max}/2$, the value will otherwise beset to $v_{max}/2$, then the algorithm can be expanded to search within the whole space to avoid the algorithm to be convergence prematurely.

4) The replaced formula of particles

Now to replace all the elements of formula 6, parameters *C1, C2,* v_{max} will be set based on the algorithm, with *n* linking pages and the maximum out-degree, the biggest out-degree, the First Degree of homepage, the iteration times *T* determined by the administrator. By referring to parameters researching results of [5], the value of *p* in formula 6 will be set to be 0.5, and make the *w* to be reduced from 1.2 to 0.4, formula 8 will be used to calculate. Using this calculation formula can make the algorithm have better space searching capabilities in the prophase, and better convergence latter, among them *i* is an integer from 0 to *T*.

$$w = \frac{0.8 \times (T - i)}{T} + 0.4 \qquad (8)$$

5) *The termination of the standard and the output of the results*

When iteration times are greater than *T*, the procedure will be stopped, then the optimum matrix and the best fitness value of the whole situation will be put out, at the mean time the running time, the setting parameters by users and the final operating results will be transferred to the log file, the log file and running program files are in the same directory, when the log file does not exist, the program will be automatically created and the relevant data will be recorded.

Experiment Results and Analysis

In order to check the performance of the model, the experimental test is conducted.

E. Test environment

The processor: the Intel Pentium (R) (R) Dual E2160; memory capacity: 2048MB; Processing system: Windows XP2.

F. Examples for computing

A simulated e-commerce website linking structure is used for test, which includes 28 pages, as shown in Fig. 1, and its additional linking structure shown in table I. Assume in the past a period of time, the time sequence visited by customers can be used to count its access frequency.

- 0-1-2-13-19-20-1-6-11-18-25
- 0-12-4-11-18-23-0-3-10-4-12-5-16-7-15-23-0-2-8-10-0
- 0-4-19-20-9-17-27-4-12-5-16-21-1-5-18-26-2-12-5-1-6-0
- 0-2-7-14-20-3-9-18-25-22-16-7-11-18-24
- 0-1-3-9-17-19-3-21-4-11-1-5-13-20-1-6-0-19-20-1-4-12-5-16-21-4-19
- 0-19-3-25-22-16-23-0-4-19-3-21-4-11-12-0-19-20-1-5-18-24-8-7-11-12-5-1-6-7-14-20
- 0-2-6-0-12-4-19-3-10-4-11-18-26-2-6-0-1-5-18-25-22-16-21-1-5-18-24-9-18-23-3-25-7-15-23-6-7-14-20

TABLE I. ADDITIONAL LINKING BEFORE OPTIMIZATION

(0,4)	(0,12)	(0,19)	(2,6)	(2,12)	(3,21)	(3,25)
(4,19)	(5,1)	(5,16)	(5,18)	(6,0)	(6,7)	(6,11)
(7,11)	(8,7)	(8,10)	(9,13)	(10,0)	(10,3)	(10,4)
(11,1)	(11,12)	(11,18)	(12,0)	(12,4)	(12,5)	(13,5)
(14,1)	(14,7)	(14,20)	(15,0)	(15,7)	(15,19)	(15,23)
(16,7)	(17,19)	(17,27)	(18,3)	(18,23)	(19,0)	(19,3)
(19,20)	(20,1)	(20,3)	(20,9)	(21,1)	(21,4)	(22,3)
(22,16)	(23,0)	(23,3)	(23,6)	(24,6)	(24,8)	(24,9)
(25,6)	(25,7)	(25,8)	(26,2)	(26,4)	(27,2)	(27,4)

G. The selection of experimental parameters

For self-adapting inertia *w*, its value will generally be taken between [0.4, 1.2][5], which can make the algorithm have better space searching capabilities in prophase, and better convergence in latter stages. As to the particles population size, the bigger the size of the population, then the more comprehensive information is gained and the better searching capability is achieved. When the particle swarm scale is 1, the algorithm is turned into an individual searching technology, then no comprehensive information can be used, and local optimum might be resulted; but when the particle swarm scale is very big, although the capability of overall optimization can be strengthened, the running time will be longer, and the convergence speed will be slowed. [5] Has studied the value setting of population scale searching, and the value 20 is considered appropriate.

Learning factor *C1*, *C2* can function as a balance of the capabilities between local searching and the overall searching. If the learning factors are too small, then the particles may be too far away from the target area, but if they are too big, it might result particles to fly to the target area suddenly or to fly out of the target area, so the appropriate value should be chosen, by referring to the researching results of [5], the appropriate value of *C1*, *C2* should be set to 2.0.

H. *The influences of iteration times on the algorithm*

Using the above set parameters, change the iteration times, and make each iteration times run for 10 times, then to calculate and record the average fitness value and the average running time of the 10 times, 7 is chosen as the biggest out- degree and the maximum value of homepage. As shown in table II, if the fitness value is changing from 0.375 to 0.379, the changing range is comparatively small, which is because at the initial process of particles, the access frequency is also included, and it makes the particle move towards the better position from very beginning. In addition, with the increase of iteration, the average running time of the system increases almost linearly. When particle swarm scale is 20 and iteration times increase every 50 times, then the additional running time value does not exceed 300ms. It can be easily seen, this algorithm does have its advantages.

TABLE II. RELATIONS BETWEEN ITERATION TIMES AND FITNESS VALUE

Iteration times	Fitness value	Running time (ms)	Iterations times	Fitness value	Running time (ms)
1	0.375	18.3	250	0.3754	1196.4
50	0.3767	283.6	300	0.3772	1422.8
100	0.3758	496.5	350	0.3756	1676
150	0.3755	738.8	400	0.3766	1902.5
200	0.3758	941.7	450	0.3771	2148

I. *Influences of the out-degree of website pages on the performance of the algorithm*

Every maximum out-degree and the biggest out-degree of homepage (as to additional linking) are run 10 times, then record the average value of the fitness, we can see that as the biggest out-degree of homepage appears, shown in table III and when maximum value of out-degree of other pages has declined, the correspondingly optimal fitness value will drop accordingly; As to the maximum out-degree of other pages, when the biggest out-degree of homepage decreases, its fitness value will reduce accordingly, but the reduction range is not as big as the reduction range of the homepage. Therefore, an out-degree of pages is an important influential factor to the fitness values. The improved fitness value can be achieved through improving the out-degree of pages, but it may lead to information overload in a single page.

Take (7,7) as the example, results of optimized website can be achieved. The additional linking besides the basic linking can be shown in table IV (Note: "+" represents the newly increased linking after the optimization; and "-" represents the deleted linking after the optimization).

In practical applications, the out-degree of each page can be set individually, so that numbers of linking pages can be accurately controlled.

TABLE III. THE RELATIONSHIP BETWEEN THE FITNESS VALUE AND THE OUT-DEGREE OF PAGES

(Out- degrees of homepage, maximum out-degrees)	Fitness value	Running time (ms)
(7,7)	0.3869	4540
(7,6)	0.3296	4556
(7,5)	0.2797	4681
(6,7)	0.376	4581
(6,6)	0.3206	4603
(6,5)	0.2698	4571
(5,7)	0.3714	4618
(5,6)	0.318	4550
(5,5)	0.2626	4581

Use Formula 3 to calculate, the original fitness value is 0.2095. As shown in Table II and III, we can see the running efficiency of the algorithm is improved, and the fitness value after optimization is higher than its original value averagely, thus the linking structure of e-commerce websites can be optimized.

Conclusions

In this paper, we abstracted the linking structure model as a mathematical model, and used PSO algorithm suitable for the optimized linking structures to solve the solution. Taking the optimal function factor into consideration in the initialization process of particles, then the original particles will be close to the optimal solution; and in the movement process of particles, make the inertial variables change linearly to improve its space searching capabilities, and meantime prevent premature convergence. Simulating test results show that the algorithm can be used to optimize e-commerce website linking structure and thus to improve the website maintenance, which has certain reference value for the application. Because of the limited size of this article, the parameter setting of particle swarm algorithm have not been thoroughly illustrated so as to improve the algorithm efficiency, and this might be the next researching focus.

References

[1] T.NAKAYAMA,H.KATO and Y.YAMANE,Discovering the gap between web site designers' expectations and users' behavior.Artificial Intelligence,2000, 245-275.

[2] N.X.Zhang,Algorithm and Datastructur-C Language Description, Higher Education Press,2006.

[3] D.W.Wang,J.W. Wang and H.F.WANG, Intelligent Optimization ,CN:Higher Education Press,2007.4

[4] J.Kennedy,R.Eberhart,A Discrete Binary Version of the Particle Swarm Algorithm. Proceedings of the World Multiconference on Systemics,Cybernetics and Informatics,1997.4104- 4109.

[5] Y.SHI,R.Eberhart.Parameter Selection in Particle Swarm Optimization.Proc of the 7th Annual Conf on Evolutionary Programming. 1998.591-600.

TABLE IV. CHANGES OF ADDITIONAL LINKING BEFORE AND AFTER THE OPTIMIZATION

+(0,5)	+(0,6)	+(0,7)	-(0,12)	-(0,19)	+(1,15)	+(1,18)
+(1,19)	+(1,23)	+(2,3)	+(2,5)	-(2,6)	-(2,12)	+(2,15)
+(2,18)	+(2,20)	+(3,0)	+(3,1)	+(3,2)	+(3,4)	+(3,5)
-(3,21)	-(3,25)	+(4,0)	+(4,1)	+(4,2)	+(4,3)	+(4,5)
-(4,19)	+(5,0)	+(5,2)	+(5,3)	+(5,4)	+(5,6)	-(5,16)
-(5,18)	+(6,1)	+(6,2)	+(6,3)	+(6,4)	+(6,5)	-(6,11)
+(7,8)	-(7,11)	+(7,12)	+(7,22)	+(7,23)	+(8,0)	+(8,1)
+(8,2)	+(8,5)	+(8,6)	+(9,1)	+(9,2)	+(9,3)	+(9,4)
-(9,13)	+(9,14)	+(10,1)	+(10,2)	+(10,5)	+(10,6)	-(11,1)
+(11,5)	+(11,6)	+(11,7)	+(11,10)	+(11,16)	+(12,1)	+(12,2)
+(12,3)	+(12,6)	+(13,0)	+(13,1)	+(13,2)	+(13,4)	-(14,1)
+(14,6)	+(14,8)	+(14,12)	+(14,18)	+(14,22)	-(15,0)	+(15,2)
……	+(26,15)	+(27,1)	+(27,5)	+(27,6)	+(27,12)	+(27,16)

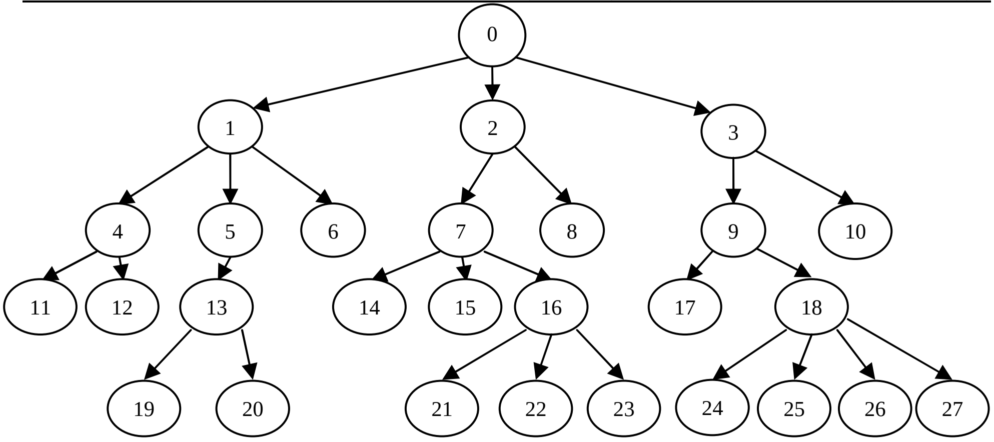

Figure 1. simulating basic linking structure