A Novel Web Optimization Technique using Enhanced Particle Swarm Optimization

P.N.Nesarajan
Research Scholar, Erode Arts & Science College, Erode

M.Venkatachalam, Ph.D
Associate Professor & HOD of Electronics, Erode Arts & Science College, Erode

T.Ranganayaki
Associate Professor in Computer Science, Erode Arts & Science College, Erode

ABSTRACT
Web performance is very important. One way to improve performance is through caching. But caching is already widely used, and studies suggest that much of the theoretically achievable performance from caching is already being realized. Caching has reduced bandwidth consumption and downloads latency. On the other hand, web-caching is heavy to enlarge further due to the developing amount of non-cacheable dynamic web-documents. Increasing the performance of web is an essential requirement, because its result in a huge increase in user supposed latency. This neat source of information establishes a basis for observations that can lead to improved overall performance for a given Web site. The main limitation focused in this method is to find out the optimal cache memory that should be keeping in order achieving maximum cost effectiveness. This method utilizes a successful Great Deluge algorithm based Particle Swarm Optimization (GDPSO) approach for achieving the best cache memory size which in turn decreases all the network cost. The investigation shows that hierarchical distributed caching can save important network cost through the use of the GDPSO algorithm.

General Terms
General classification of the Paper is optimization the Web by the ANN, PSO, GDPSO Algorithm

Keywords
Website optimization techniques, Web performance, Particle Swarm Optimization (PSO), Great Deluge Particle Swarm Optimization (GDPSO)

1. INTRODUCTION
Nowadays, there is very significant constraint in the development of web performance. Because the Web's popularity outcome heavy traffic in the Internet, the net effect of this growth was a significant increase in the user perceived latency. Potential sources of latency are the Web servers' low bandwidth, network congestion, bandwidth underutilization, propagation delay and heavy load. The performance of client downloads times and diminishes network traffic by caching commonly done copies of web object near to the client is improved in web caching technology. There is some limitation in web caching, there are Cache consistency (how to keep the cache copies consistent), Cache placement (where to cache copies of objects) and Client redirection (how to redirect client to the optimal cache server).

The performance of web-caches for diminishing bandwidth consumption and download latency has been very successful in the past. However, conventional web-caching is relevant to static documents, or to documents that alter in huge timescales. Since the quantity of dynamic and versus static document is enhancing day by day, present caching outcome has attained a point where their operation does not significantly enhanced unless they incorporate a mechanism to "cache" dynamic documents. The main aim of a cache is to increase the speed of computation by exploiting frequent, current or costly data used. In this work propose and calculate a static cache that works concurrently as list and intersection cache, offering a more efficient way of handling cache space. In this method, in order to offer improved performance to the PSO, Great Deluge (GD) algorithm based PSO is used for providing successful optimized results.

The review of this research is organized as follows. Section 2 summarizes the concepts and literature survey. Section 3 discusses the proposed method, and section 4 provides the experiments with high accuracy. Finally, Section 5 presents the conclusions of the work.

2. LITERATURE SURVEY
In highly competitive manufacturing industries nowadays, the manufactures ultimate goals are to produce high quality product with less cost and time constraints. To achieve these goals, one of the considerations is by optimizing the machining process parameters such as the cutting speed, depth of cut, radial rake angle are given by [15]. The measure of similarity between objects is a very useful tool in many areas of computer science, including information retrieval. [9] present a technique to estimate the accuracy of computing SimRank iteratively. This technique provides a way to find out the number of iterations required to achieve a desired accuracy when computing SimRank.

[3] Presents a review of the current state of the art in computational optimization methods applied to renewable and sustainable energy, offering a clear vision of the latest research advances in this field. In wireless sensor networks energy consumption is one of the biggest constraints of the wireless sensor node and this limitation combined with a typical deployment of large number of nodes has added many challenges to the design and management of wireless sensor networks are given by [12]. [5] proposed a method that substantially increases the conversion efficiency at light loads by minimizing switching and driving losses of semiconductor switches, as well as core losses of magnetic components.

Map Reduce jobs are amenable to many traditional database query optimizations, but existing systems do not apply them, substantially because free-form user code obscures the true data operation being performed by [4]. [14] investigates the feature subset selection problem for the binary classification problem using logistic regression model. They developed a modified discrete particle swarm optimization (PSO) algorithm for the feature subset selection problem. The particle swarm is a population-based stochastic algorithm for optimization which is based on social–psychological principles. [8] Provide interactions result in iterative improvement of the quality of problem solutions over time.

Particle swarm optimization (PSO) has undergone many changes since its introduction given by [11]. This paper comprises a snapshot of particle swarming from the authors' perspective, including variations in the algorithm, current and ongoing research, applications and open problems. The improvisation is preformed through moving the particles around the search space by means of a set of simple mathematical expressions which model some inter-particle communications are given by [7].

[10] Carried out a set of simulation experiments to test the proposed model when applied to a Muskingum model, and we compared the results with eight superior methods. [6] the PSO variants, devised for dynamic optimization problems, are reviewed. This is the first comprehensive review that is conducted on PSO variants in dynamic environments. [16] proposed an algorithm that takes a particle swarm optimization (PSO) as the main evolution method.

A two-stage memory architecture and search operators exploiting the accumulated experience in memory are maintained within the framework of a Great DeLuge algorithm for real-valued global optimization is introduced by [2]. The level-based acceptance criterion of the Great DeLuge algorithm is applied for each best solution extracted in a particular iteration. The use of memory-based search supported by effective move operators results in a powerful optimization algorithm. [1] proposed hybrid algorithm has been extensively compared with the original BFOA algorithm and the PSO algorithm. Simulation results have shown the validity of the proposed BSO in tuning SVC compared with BFOA and PSO.

3. RESEARCH METHODOLOGY

In this approach, in order to provide better performance to the ANN and PSO, Great Deluge (GD) algorithm based PSO has been utilized for providing effective optimized results.

3.1 Artificial Neural Networks (ANNs)

The Back propagation Algorithm (BP) is a standard domain-dependent method for supervised training (Figure 1). This work is performing is done by calculating the output error, measuring the gradient of this error, and alter the ANN weight in the descending gradient direction.

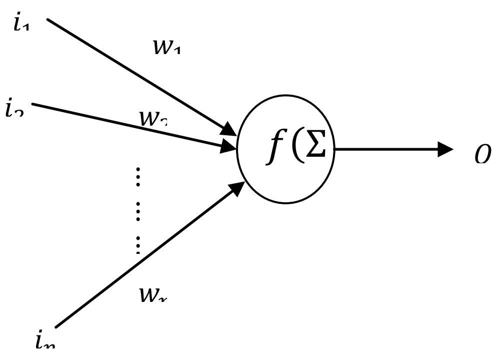

Figure 1: ANN Neuron

Therefore, BP is a gradient-descent local search procedure. The squared error of the ANN for a set of patterns is:

$$E = \sum_{p=1}^{m} \sum_{i=1}^{n} \left(t_i^p - o_i^p\right)^2 \quad (1)$$

The definite value of the preceding expression related on the weights of the network. The standard BP algorithm calculates the gradient of E (for all the patterns) and updates the weights by moving them along the gradient-descendent direction. This can be summarized with the expression $\Delta w = -\eta \nabla E$, where the parameter $\eta > 0$ is the learning rate that manage the learning process. The pseudo-code of the BP algorithm is given below.

Initialize Weights,

While not Stop-Criterion do

For all i, j do

$$w_{ij} = w_{ij} - \eta \frac{\partial E}{\partial w_{ij}}$$

End for

End While

Pseudo code of back propagation algorithm

3.2 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) technique for optimization was introduced by Sulaiman et al (2008) and is motivated by the developing action of a collection of birds look for food. This optimization tool is based on population; to overcome from several optimization problems this tool is easily applied and implemented. It is utilized to investigate the look for space of a specific problem to find the settings or parameters need to increase a specific objective. Fast convergence is a main strength for PSO, which evaluate favorably with several global optimization algorithm such as Simulated Annealing (SA), Genetic Algorithm (GA) and additional global optimization algorithm.

Every particle understands its great value (pbest) and its location. This information is a comparison of individual experience of every particle. Additionally, all particles know great value so far in the clustering between pbests. Based on current velocities, distance between the current position, current position, gbest and pbest all particles try to modify its direction.

The PSO algorithm consists of three steps, which are repeated until stopping criteria is met:

Step 1: Evaluate the fitness of each particle

Step 2: Update personal best (pbest) of each particle, and global best (gbest)

Step 3: Update velocity and position of each particle

Fitness calculation is performed by supplying the candidate result to the objective function or fitness function. Personal best (pbest) of each particle, global best(gbest) and directions are updated by evaluating the newly performed fitness beside the earlier individual's pbest and gbest .

The velocity and direction update procedure is responsible for the optimization capacity of the PSO algorithm. The velocity of each particle is updated using the following equation:

$$V_i(t+1) = W^* V_i(t) + c_1 r_1 [pbest(x(t)) - x_i(t)] + c_2 r_2 [gbest(t) - x_i(t)] \quad (2)$$

$$x_i(t+1) = x_i(t) + V_i(t+1) \quad (3)$$

The particle of index is represented by i. The velocity of particle i at time t is denoted as vi(t) and the position of particle i at time t is denoted as xi(t). The user supplied coefficients use parameters are w, c1, and c2. Velocity update use random values of r1 and r2. Individual best candidate solution of pbest(x(t)) used for particle i at time t, and global best candidate of g(t) used at time t. The inertia weight is denoted as w, then influence of social and cognitive components are determined by c1 and c2. Equation 3 is used to update the position of the particle. This process is repeated until the best solution is found or terminate conditions are satisfied.

3.3 Great Deluge based Particle Swarm Optimization (GDPSO)

This method offers an improved version of Particle Swarm Optimization algorithm using Great Deluge algorithm called GDPSO. The achieved outcome is compared with the best solution and it will be accepted because of better performance, after receiving a new result in standard PSO. The best identified solution and parameter of "Water Level" or WL is used to compare with the proposed approach result. If proposed approach is better and accurate then both, it is accepted as a new solution.

The proposed algorithm is essentially dissimilar with the basic PSO so that it tries to use the standard method of Great Deluge local search in the PSO algorithm. The acceptance level is denoted by WL parameter and to find the permissible range of answers the parameter of UP is used. The performance of the UP parameter is to minimize or maximize the WL.A new technique is performed on particular standard function and evaluation of the proposed algorithm is compared with standard PSO. Thus the proposed GDPSO algorithm offers better accurate result with less convergence rate and it escape from the local optimum.

```
For each particle i
   Randomly initialize v_i x_i = p_i
Evaluate f(p_i)
                 p_g = argmax{f(p_i)}
End for
Choose WL and UP
Repeat
Each particle i
Update particle position x_i according to
equation below
       v_i = x[v_i + c_1e_1 − (p_g − x_i) + c_2e_2
                        − (p_i − x_i)].
                 x_i = x_i + v_i
Evaluate f(x_i)
                 if (f(x_i) > f(p_g))
End if
     if ((f(xi) > f(pi))(&&f(x_i) > WL))
                 p_g = argmax{f(p_i)}
End if
                 WL = WL + Up
Until termination criterion reached
```

Pseudo-code of GDPSO

4. EXPERIMENTAL RESULTS

In this section, experimental result is performed by using MATLAB. Then perform a comparison between standard Web cache, ANN Web cache, PSO Web cache and GDPSO Web cache. The web server gives request to standard web caches, those requested information is stored locally, and then transfer the information to the client. If the Web cache receives a request for the related information for the next time, it basically returns the locally cached data instead of searching all over the Internet. On the other hand, ANN and PSO Web cache request from the Web server and determine which request should be stored locally using proposed approach.

In this paper, the number of hidden nodes is determined by using 2n+1. The number of output nodes is relatively easy to specify as it is directly related to the undertaken problem. In this study, only one output node is needed; about the decision to cache or not to cache the data. GDPSO training parameters for the Web caching is set as follow:

Learning rate = 0.7

Total error = 0.003

Individual error = 0.003

Number of epoch = 30

Number of hidden layer = 1

Number of nodes in hidden layer = 7

Stopping condition = total error reached or maximum number of epoch

The pre-processing is the key component in Web cache. At this stage, three attributes are proposed, which are based on the attributes that are widely used by the researchers in the area of Web performance analysis. The attributes used in this study are:

1. Time: Time is the counter that observes the time takes to receive a data. The time stated in seconds.
2. Script Size: The size of the data that is fetched. The size is expressed in bytes and kilobytes.
3. Numbers of Hit: Observing the number of hits per data. Where on each request done for a Web file, the Number of Hit counter for requested file will be increased.
4. Hit rate: The ratio of requests fulfilled by the cache, and then not handled by the origin services. It gives a rough estimation of both the saved network traffic and the reduction of latency perceived by the client.
5. Weighted hit rate: the ratio of bytes served to the client by the cache.
6. Latency: the time that an end-user waits for retrieving a resource. It is usually a short latency is simple desirable, for strongly time-related resources that become a strict requirement.

Each attribute must be multiplied with defined Priority Value (PV) to get the total of the attributes for target output generation of the network. An example is shown as:

Expected target = (size *0.466255) + (hit *0.398721) +(time *0.327787)+(hit rate *0.0.267463) + (weight *0.2071771) +(latency *0.167883)

In this section, we present a performance comparison between standard ANN Web cache, BPN and PSO Web cache. The standard Web cache fills requests from the Web server, stores the requested information locally, and sends the information to the client. If the Web cache gets a request for the similar information for the next time, it simply returns the locally cached data instead of searching over the Internet. On the other hand, ANN and GDPSO Web cache request from the Web server and determine which request should be stored locally using AI approach.

The testing log data are collected from college, which were running at the Bharathiar University. Six different parameters are chosen for training; size, retrieval time, number of hit rate, weight and latency. The table 1 shows the performance of Error, Total Iteration and Accuracy Test for the month of September, October and November 2014.

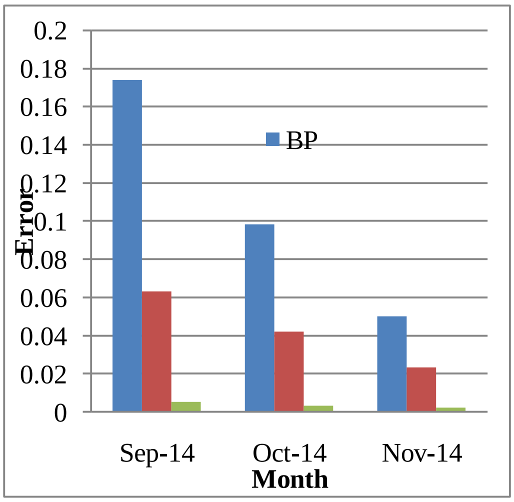

Figure 3: Mean square error

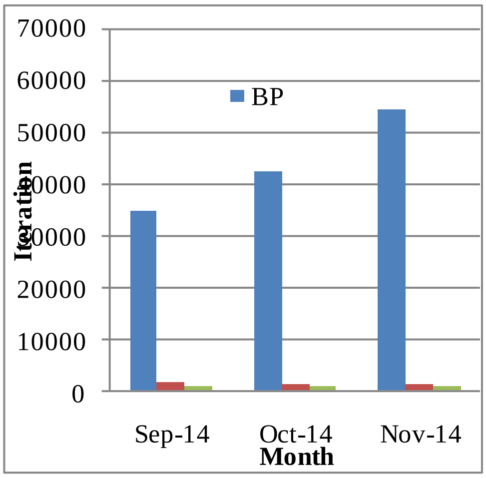

Figure 4: Total training iteration

Figure 3 and figure 4 show the mean square error and total training iteration for BP, PSO and GDPSO. The proposed method of GDPSO has less error rate and less training iterative. Figure 5 shows the accuracy test, GDPSO has high accuracy when compare with BP and GDPAO.

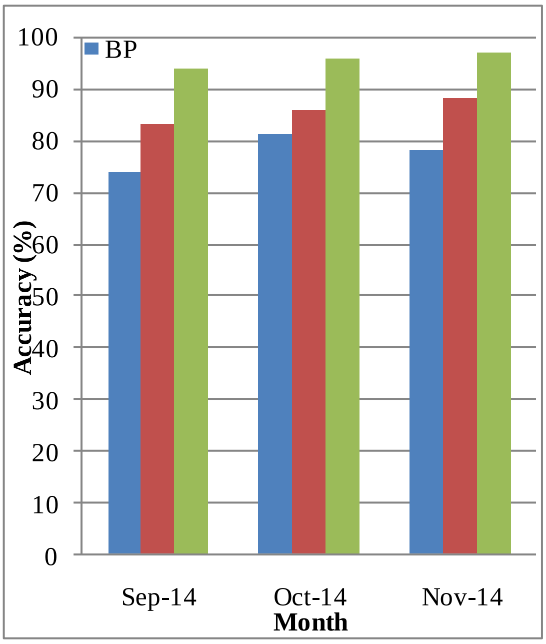

Figure 5: Accuracy test

5. CONCLUSION

Web caching is an effective solution to learn about web service, it used to improve scalability of web system and to minimize the traffic over the internet. The Web caching can complement each other since the web caching exploits the temporal locality for predicting revisiting requested objects. Thus, combination of the web caching doubles the performance compared to single caching. Web cache algorithm in order to reduce the cache misses and increases the hit ratio. The various design issues of Web caching algorithms such as load balancing, cache miss, transparency, scalability and cache coherence are analyzed. This paper reviews principles and some the existing web caching approach.

6. REFERENCES

[1] Abd-Elazim, S. M., and Ali, E. S. 2014. A hybrid particle swarm optimization and bacterial foraging for power system stability enhancement. Complexity.

[2] Acan, A., and Ünveren, A. 2014. A two-stage memory powered Great Deluge algorithm for global optimization. Soft Computing, 1-21

[3] Banos, R., Manzano-Agugliaro, F., Montoya, F. G., Gil, C., Alcayde, A., and Gómez, J. 2011. Optimization methods applied to renewable and sustainable energy: A review. Renewable and Sustainable Energy Reviews, 15(4), 1753-1766.

[4] Jahani, E., Cafarella, M. J., and Ré, C. 2011. Automatic optimization for MapReduce programs. Proceedings of the VLDB Endowment, 4(6), 385-396.

[5] Jang, Y., and Jovanovic, M. M. 2010. Light-load efficiency optimization method.Power Electronics, IEEE Transactions on, 25(1), 67-74.

[6] Jordehi, A. R. 2014. Particle swarm optimisation for dynamic optimisation problems: a review. Neural Computing and Applications, 25(7-8), 1507-1516.

[7] Kaveh, A. 2014. Particle Swarm Optimization. In Advances in Metaheuristic Algorithms for Optimal

Design of Structures (pp. 9-40). Springer International Publishing.

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

[9] Lizorkin, D., Velikhov, P., Grinev, M., and Turdakov, D. 2010. Accuracy estimate and optimization techniques for SimRank computation. The VLDB Journal—The International Journal on Very Large Data Bases, 19(1), 45-66.

[10] Ouyang, A., Li, K., Truong, T. K., Sallam, A., and Sha, E. H. M. 2014. Hybrid particle swarm optimization for parameter estimation of Muskingum model.Neural Computing and Applications, 25(7-8), 1785-1799.

[11] Poli, R., Kennedy, J., and Blackwell, T. 2007. Particle swarm optimization. Swarm intelligence, 1(1), 33-57.

[12] Sendra Compte, S., Lloret, J., García Pineda, M., and Toledo Alarcón, J. F. 2011. Power saving and energy optimization techniques for Wireless Sensor Networks. Journal of communications, 6(6), 439-459.

[13] Sulaiman, S., Shamsuddin, S. M., Forkan, F., and Abraham, A. 2008, May. Intelligent Web caching using neurocomputing and particle swarm optimization algorithm. In Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on (pp. 642-647). IEEE.

[14] Unler, A., and Murat, A. 2010. A discrete particle swarm optimization method for feature selection in binary classification problems. European Journal of Operational Research, 206(3), 528-539.

[15] Yusup, N., Zain, A. M., and Hashim, S. Z. M. 2012. Evolutionary techniques in optimizing machining parameters: Review and recent applications (2007–2011).Expert Systems with Applications, 39(10), 9909-9927.

[16] Zhang, Z., Jiang, Y., Zhang, S., Geng, S., Wang, H., and Sang, G. 2014. An adaptive particle swarm optimization algorithm for reservoir operation optimization. Applied Soft Computing, 18, 167-177.