A FRAMEWORK OF ROUGH CULTURAL ALGORITHMS IN OPTIMIZING MOBILE WEB CACHING PERFORMANCE

Sarina Sulaiman[1], Siti Mariyam Shamsuddin[2], Ajith Abraham[3], Shahida Sulaiman[4]

[1,2] Soft Computing Research Group, Faculty of Computer Science and Information System,
Universiti Teknologi Malaysia,
81310 Skudai, Johor, Malaysia.

[3]Centre for Quantifiable Quality of Service in Communication Systems,
Centre of Excellence,
Norwegian University of Science and Technology,
O.S. Bragstads plass 2E,
N-7491 Trondheim, Norway.

[4]School of Computer Sciences, Universiti Sains Malaysia,
11800 USM, Penang, Malaysia.

email : sarina@utm.my[1], mariyam@utm.my[2], ajith.abraham@ieee.org[3], shahida@cs.usm.my[4]

Abstract: The stipulation of internet content rises dramatically in recent years. Servers have become extremely powerful and the bandwidth of end user connections and backbones grow constantly during the previous decade. Nonetheless, users frequently experience poor performance in accessing web sites or downloading files primarily when they use mobile devices due to their limited storage, processing, display, power and communication resources.. The common cause of poor performance is due to the direct access to the server (e.g. pitiable performance of server-side applications or during burst crowds) and network infrastructure (e.g. long geographical distances, network overloads, etc.). Hence, the goal of this study is to propose Rough Set (RS) as a knowledge representation for uncertainty in data of client behavior and mobile event specification with resource dependencies to reduce latency by prefetching selected resources to resolve the problems in handling dynamic web pages. Simultaneously, Cultural Algorithms (CA) will be exploited to optimize the performance of proxy caching in accumulating the knowledge between each generation. Consequently, The proposed caching scheme, Rough Cultural Algorithm (RCA) is developed to materialise the caching policies.

Keywords: Mobile Web Caching, Rough Set, Cultural Algorithms, Prefetching, Proxy Caching.

1. Introduction

Caching is a technique used to store popular documents closer to the user. It uses algorithms to predict user's needs to specific documents and stores important documents. According to Curran and Duffy[1], caching can occur anywhere within a network, on the user's computer or mobile devices, at a server, or at an Internet Service Provider (ISP). Many companies employ web proxy caches to display frequently accessed pages to their employees, as such to reduce the bandwidth with lower costs [1][2]. Web cache performance is directly proportional to the size of the client community [3][1].The bigger the client community, the greater the possibility of cached data being requested, hence, the better the cache's performance [1].

Caching a document can also cause other problems. Most documents on the Internet change over time as they are updated. Static and Dynamic Caching are two different technologies that widely used to reduce download time and congestion [1]. *Static Caching* stores the content of a webpage which does not change. There is no need to request the same information repeatedly. This is an excellent approach to fight congestion. *Dynamic Caching* is slightly different. It determines whether the content of a page has been changed. If the contents have changed, it will store the updated version [4]. This unfortunately can lead to congestion and thus it is possibly not a very good approach as it does require verification on the source of the data prior to updating. If these two technologies are implemented simultaneously, then the latency and congestion can be diminished.

Prefetching is an intelligent technique used to reduce perceived congestion, and to predict the subsequently page or document to be accessed [5]. For example, if a user is on a page with many links, the prefetching algorithm will predict that the user may want to view associated links within that page. The prefetcher will then appeal the predicted pages, and stores them until the actual request is employed. This approach will display the page significantly faster compared to page request without prefetching. The only drawback is that if the user does not request the pages, the prefetching algorithm will still implement the prediction of the subsequent pages, thus causes the network to be congested.

In this study, we proposed mobile Web caching scheme with an integration of soft computing technique. Rough Set (RS) can help to select relevant features of client behavior and mobile event specification to prefetch selected resources in handling dynamic web pages. Meanwhile, Cultural Algorithms (CA) will be proposed to optimize the performance of proxy caching that will store the knowledge between each generation. The paper is structured as follows. A literature review is presented in Section 2 that describes on mobile Web caching, RS and CA. In Section 3, we illustrate our proposed method with Rough Cultural Algorithm (RCA), and finally, Section 4 gives the concluding remark of our study.

2. Literature Review

Section 2 describes related works on mobile caching, as well as a discussion on RS and CA techniques in optimizing the performance of web caching.

2.1 Mobile Web Caching

Caching is the most relevant techniques to improve storage system, network, and device performance. In mobile environments, caching can contribute to a greater reduction in the constraint of utilization resources such as network bandwidth, power, and allow disconnected operation [6]. A lot of studies are focus on developing a better caching algorithm to improve the choice of item to replace, and simultaneously, building up techniques to model access behavior and prefetch data. From 1990's until today, researchers on caching have produced different caching policies to optimize a specific performance and to automate policy parameter tuning. Prior to this, administrator or programmer had to select a particular parameter to observe workload changes. However, an adaptive and self-optimizing caching algorithms offer another advantage when considered mobile environments, where users of mobile devices should not expect to tune their devices to response the workload changes [6]. The workload depends on the current position of the mobile node in relation to other nodes and stations, and also depends on the current location and context of the mobile user.

Caching is effectively for data with infrequent changes. In addition, caching data locally to mobile nodes helps the ability to retrieve data from a nearby node, rather than from a more distant base station [6]. By simply retrieving data using multiple short-range transmissions in wireless environments provides a reduction in overall energy consumed. Santhanakrishnan et al. [6] illustrated on the demand-based retrieval of the web documents in the mobile web. They proposed caching scheme; Universal Mobile Caching which performed the most basic and general form of caching algorithms and largely emphasize the impact of the adaptive policy. This scheme is suitable for managing object caches in structurally varying environments. Ari et al. [7] proposed Adaptive Caching using Multiple Experts (ACME) which individual experts were full replacement algorithms, applied to virtual caches, and their performance was estimated based on the observed performance of the virtual caches.

Wu et al. [8] introduced a rule-based modular framework for building self-adaptive applications in mobile environments. They developed techniques that combine static and dynamic analysis to uncover phase structure and data access semantics of a rule program. The semantic information is used to facilitate intelligent caching and prefetching for conserving limited bandwidth and reducing rule processing cost. As well, Komninos and Dunlop [9] found that calendars can really provide information that can be used to prefetch useful Internet content for mobile users. While it is expected that such an approach cannot fulfill the whole of Internet content needs for a user, the work presented here provides evidence to the extent to which a mobile cache can be populated with relevant documents that the user could find of interest. However, a foreseeable problem with the current system is that the current adaptation algorithm adjusts the system gradually, and not immediately, to the needs of a user. Thus, if a dramatic change of circumstances was to occur, or if a user was to require information from a very specific and known source, it is likely the system would fail to provide the necessary information.

2.2 Mobile Event Specification

The events in a mobile information system can be any changes to the system itself or the way it is being used. Events can be primitive or composite [8]. A primitive event is to model a certain level of change on a single source (such as disk capacity, free memory, bandwidth, etc.). Primitive events can be combined using event operators to form composite events. Wu et al. [8] classified the events of interest in a mobile environment into resource events, mobility events, and environment events as depicted in Figure 1.

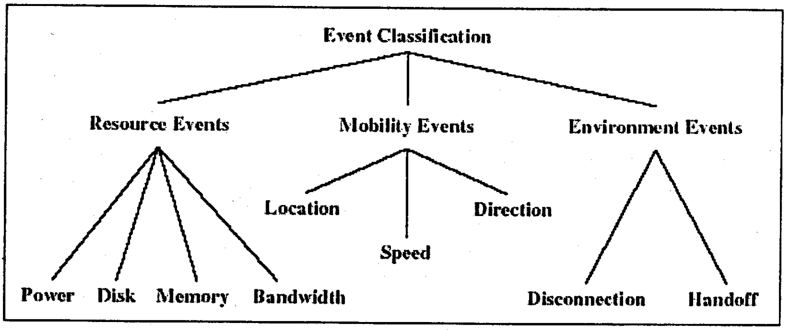

Figure 1: Event Classification [8].

According to Wu et al. [8], resource events model the change in resource availability on a mobile host. An application must be aware of the current resource status in order to better utilize the usually limited resources. Thus, they have identified the available disk space, free memory, battery power, and the wireless bandwidth level as the primary resources for monitoring and detection. Other resource types can be easily added in similar way. Mobility events [8] are related to the change in position of the mobile host under consideration. This is a unique characteristic in mobile computing which has no direct counterpart in traditional distributed systems. They proposed to monitor the current position, speed, and direction as three key parameters for representing the mobile host's current mobility status. As a first step, they used the current cell ID within which the mobile host resides as its current position.

Environment events [8] are changes to the operating environment that are expected to have significant effect on the mobile host. Wu and friends [8] identified disconnection and handoff as the two most distinctive state changes to monitor for the mobile client. This allows applications to take preventive actions in response to these changes.

2.3 Performance Analysis with Integration of Soft Computing and Evolutionary Computation Techniques

Population-based heuristic methods are iterative solution techniques that handle a population of individuals which are evolving according to a given search strategy. At each iteration, periods of self-adaptation (mutations) alternate with periods of cooperation (crossover), and periods of competition (selection). The population-based heuristic search is dependent of the following components: the knowledge representation for the specific problem we want to solve and the search strategy or the evolution process [11]. The adaptability of an individual represents its ability to survive in an uncertain environment [10]. Artificial Intelligence researchers have explored different ways to represent uncertainty: belief networks, default reasoning, Dempster-Shafer theory, Fuzzy set theory, Rough Set Theory (RST) [12]. The RST with the three-valued simplicity, lower, upper, and boundary approximation sets, works well on discrete and categorical data. RS can be useful even with missing data, changes of scale, and problems where membership grades are hard to define, and problems requiring changes in the partition [20].

2.3.1 Uncertainty

Uncertainty, as well as evolution, is a part of nature. When humans describe complex environments, they use linguistic descriptors of cognized real-world circumstances that are often not precise, but rather "Fuzzy". The theory of fuzzy sets provides an effective method of describing the behavior of a system, which is too complex to be handled with the classical precise mathematical analysis [13]. The theory of RS emerged as another mathematical approach for dealing with uncertainty that arises from inexact, noisy or incomplete information [14]. Fuzzy set theory assumes that the membership of the objects in some set is defined as a degree ranging over the interval [0,1]. RST focuses on the ambiguity caused by the limited distinction between objects in a given domain.

Uncertainty occurs in many real-life problems. It can cause the information used for problem solving being unavailable, incomplete, imprecise, unrealiable, contradictory, and changing [24]. In computerized system, uncertainty is frequently managed by using quantitative approaches that are computationally intensive. For example, a binary that processes 'TRUE or FALSE', or 'YES' or 'NO' type of decisions, is likely to arrive at a conclusion or a solution faster than one that needs to handle uncertainty.

Organizing uncertainty is a big challenge to knowledge-processing systems [24]. In some problems, uncertainty can possibly be neglected, though at the risk of compromising the performance of a decision support system. However, in most cases, the management of uncertainty becomes necessary because of critical system requirements or more complete rules are needed. In these cases, eliminating inconsistent or incomplete information when extracting knowledge from an information system may introduce inaccurate or even false results, especially when the available source information is limited. Ordinarily, the nature of uncertainty comes from the following three sources: inconsistent data, incomplete data and noisy data.

2.3.2 Rough Set (RS)

Another approach to represent uncertainty is with RS. RS are based on equivalence relations and set approximations, and the algorithms for computing RS properties are combinatorial in nature. The main advantages of RST are as follows [24]:

- It does not need any preliminary or additional information about data;
- It is easy to handle mathematically;
- Its algorithms are relatively simple.

Wakaki et al. [19] used the combination of the RS-aided feature selection method and the support vector machine with the linear kernel in classifying Web pages into multiple categories. The proposed method gave acceptable accuracy and high dimensionality reduction without prior searching of better feature selection. Liang et al. [21] used RS and RS based inductive learning to assist students and instructors with WebCT learning. Decision rules were obtained using RS based inductive learning to give the reasons for the student failure. Consequently, RS based WebCT Learning improves the state-of-the-art of Web learning by providing virtual student/teacher feedback and making the WebCT system much more powerful. Ngo and Nguyen [22] proposed an approach to search results clustering based on tolerance RS model following the work on document clustering. The application of tolerance RS model in document clustering was proposed as a way to enrich document and cluster representation to increase clustering performance.

In general, the basic problems in data analysis that can be tackled using a RS approach are as follows [24]:

- Characterization of a set of objects in terms of attribute values;
- Finding the dependencies (total or partial) between attributes;

- Reduction of superfluous attributes (data);
- Finding the most significant attributes;
- Generation of decision rules.

2.3.3 Evolutionary Computation

Evolution can be defined in one word, adaptation in an uncertain environment [20]. Nature has a robust way of dealing with the adaptation of organisms to all kind of changes and to evolve successful organisms. According to the principles of natural selection, the organisms that have a good performance in a given environment survive and reproduce, whereas the others die off [20]. After reproduction, a new generation of offspring, derived from the members of the previous generation is formed. The selection of parents from these offspring is often based upon fitness. Changes in the environment will affect the population of organisms through the random mutations. Mayr said: "Evolution is a dynamic, two-step process of random variation and selection" [16]. Evolutionary algorithms are heuristic optimization methods inspired from natural evolution processes. According to Goldberg, there are three basic population-only mechanisms that model evolution: genetic algorithms, evolutionary strategies and evolutionary programming [17]. Each one of the methods models the evolution of a population of individuals at a different scale and applies selection and reproduction operators to find an individual that is fit with regard of the fitness function.

2.3.4 Cultural Algorithm (CA)

Edward B. Tylor was the first to introduce the term "Culture" in his two volume book on *Primitive Culture* in 1881. He described culture as "that complex whole which includes knowledge, belief, art, morals, customs, and any other capabilities and habits acquired by man as a member of society".

A Cultural Algorithm is an evolutionary computational approach that utilizes culture as a vehicle for storing relevant information that is accessible to all members of the population over the course of many generations. In this context, culture can be viewed as an evolving source of data that influences the patterns of behavior that are practiced by various members of the population. As in human societies, culture changes over time, but it provides a baseline for interpreting and documenting an individual's behavior within a society [18][24]. CA was developed to model the evolution of the cultural component over time as it learns and acquires knowledge. CA can be viewed as an extension of Genetic Algorithms, where the Belief Space acts as a medium of knowledge between each generation.

Reynolds [18] described that the nature of social intelligence is in terms of group problem solving. Furthermore, in terms of social interaction scale, the CA is suitable for global searching and compatible with other soft computing techniques. Each individual selects a suitable knowledge model based upon the experience of those who used the model previously. In optimization problem, movement of an individual is controlled by a knowledge model. Individuals communicate only indirectly through updates on knowledge model that through them. Roles emerge when an individual uses the same model repeatedly. Furthermore, CA is a class of model derived from cultural evolution process. At the micro-evolutionary level there is a population of individuals, each described in terms of a set of behavioral traits. Traits can be modified and exchanged between individuals by means of a variety of socially motivated operators. At the macro evolutionary level, individual experiences are collected, merged, generalized, and specialized in the Belief Space. This information can serve to direct the future actions of the population and its individuals. Therefore, CA is useful in exploring large search space accumulating global knowledge about the problem space. Also it is clear that cultural evolution proceeds at a faster rate than biological evolution [18][24]. Figure 2 gives a pseudo code of the Cultural Algorithm.

```
Begin Cultural Algorithm
        t = 0;
        Initialize Population POP(t);
        Initialize Beliefspace BLF(t);
        Evaluate Population POP(t);
        repeat
                Communicate (POP(t), BLF(t));
                Adjust Beliefspace BLF(t);
                Communicate (BLF(t), POP(t));
                t = t + 1;
                Select POP(t) from POP(t-1);
                Evolve POP(t);
                Evaluate Population POP(t);
        until (termination condition reached)
End Cultural Algorithm
```

Figure 2: Cultural Algorithm Pseudo Code [15][24].

The variable *t* represents the time period in terms of population generations. POP is the Population Space and BLF is the Belief Space [24].

We can view the cultural evolution process as a vehicle for amplifying individual or group behavior and building consensus. In other words during cultural evolution, 'conceptual beacons' that symbolize acceptable and unacceptable behavior for individuals in a population (society) are accumulated [23]. Individuals are first evaluated using a performance function. The performance information represents the problem solving experience of an individual. An acceptance function determines which individuals in the current population are able to impact the current beliefs. The experience of these selected individuals is merged or adjusted with those of other individuals to form group beliefs. These group beliefs are then used to guide the changes to the population at the next step.

2.3.5 Components of a Cultural Algorithm

Cultural Algorithm can be viewed as a dual inheritance system with evolution taking place at the population level and at the belief level [23]. The two components interact through a communications protocol. The protocol determines the set of 'acceptable' individuals that are able to update the Belief Space. Likewise the protocol determines how the updated beliefs are able to impact the adaptation of the population component. A component level description of the Cultural Algorithm is given in figure 3.

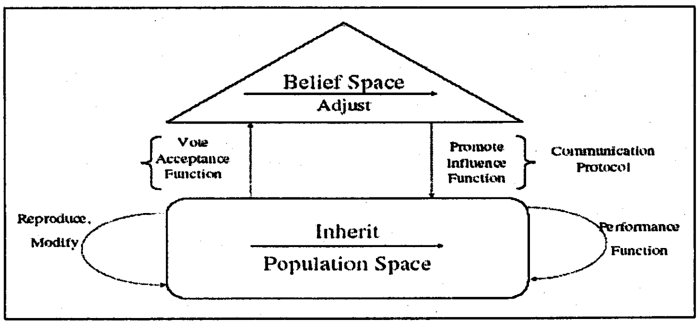

Figure 3: Components of A Cultural Algorithm [15][24].

These are the components that are used to support the definition of CA Toolkit (CAT) within the testbed that is used to configure different CA for the solution of non-linear optimization problems [15]. This testbed supports those population models that are often used for real-valued function optimization. They include real-valued genetic algorithms (GENOCOP) [15], evolutionary programming, and evolution strategies. The knowledge in the Belief Space will correspond to that information needed to reason about the problem constraints. That knowledge is expressed in the form of constraint networks as employed in the constraint satisfaction literature [24].

General features of CA as follows [18]:

- Dual Inheritance (at population and knowledge levels)
- Knowledge are "beacons" that guide evolution of the population
- Supports hierarchical structuring of population and belief spaces.
- Domain knowledge separated from individuals (e.g. ontologies)
- Supports self adaptation at various levels
- Evolution can take place at different rates at different levels ('Culture evolves 10 times faster than the biological component').
- Supports hybrid approaches to problem solving.

- A computational framework within which many all of the different models of cultural change can be expressed.

CA is suitable for problems such as [18]:

- Significant amount of domain knowledge (e.g. constrained optimization problems).
- Complex systems where adaptation can take place at various levels at various rates in the population and belief space.
- Knowledge is in different forms and needs to be reasoned about in different ways.
- Hybrid systems that require a combination of search and knowledge based frameworks.
- Problem solution requires multiple populations and multiple belief spaces and their interaction.
- Hierarchically structured problem environments where hierarchically structured population and knowledge elements can emerge.

2.3.6 A Framework of Rough Set and Cultural Algorithm

Traditionally, RS and CA have their on framework. The framework of RS tends to classification, whilst, CA framework inclines to understand the nature of social intelligence in terms of group problem solving.

a) A Framework of Rough Set

The RClass system integrates RST with an ID3-like learning algorithm [24] as shown in Figure 4. It includes three main modules; a consistency analyzer, a rough classifier and an induction engine. The consistency analyzer analyses the training data and performs two tasks; elimination of redundant data items, and identification of conflicting training data. The rough classifier has two approximators; the upper approximator and the lower approximator. The rough classifier is employed to treat inconsistent training data. The induction engine module has an ID3-like learning algorithm based on the minimum-entropy principle. The concept of entropy is used to measure how informative an attribute is.

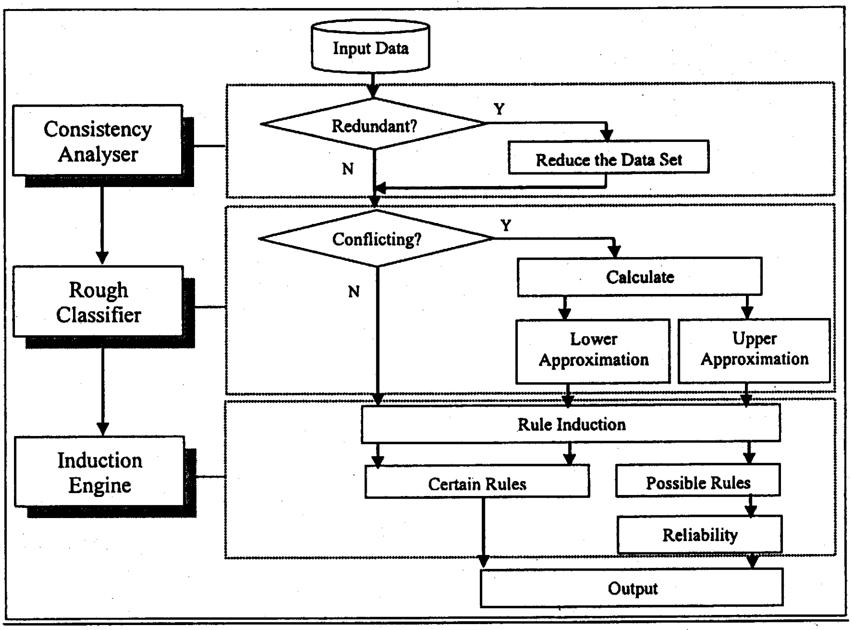

Figure 4: Framework of the RClass System [24].

b) A Framework of Cultural Algorithm

As a dual inheritance system, Cultural Algorithm has two basic components [25]: Population Space and Belief Space. First, individuals in the Population Space are evaluated with a performance function *obj()*. An acceptance function *accept()* will then determine which individuals are to impact the Belief Space. Experiences of those chosen elites will be used to update the knowledge / beliefs of the Belief Space via function *update()*, which represents the evolution of beliefs. Next, the beliefs are used to influence the evolution of the population. New individuals are *generated* under the *influence* of the beliefs, and from then, together with old individuals, individuals are *selected* and form a new generation of population. The two feedback paths of information, one through the *accept()* and *influence()* functions, and the other through individual experience and the *obj()* function create a system of dual inheritance of both population and belief. The population component and the Belief Space interact with and support each other, in a manner analogous to the evolution of human culture (Figure 5).

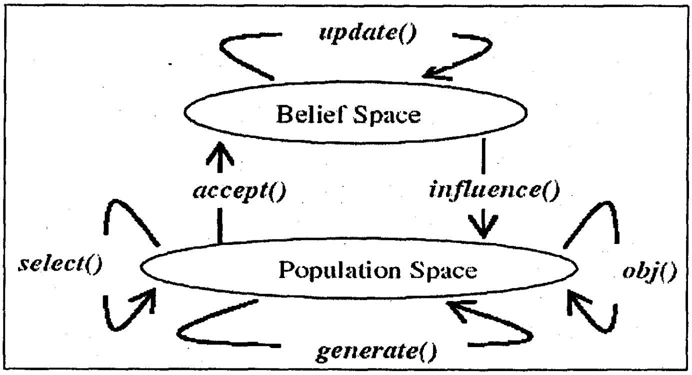

Figure 5: Framework of Cultural Algorithm [25].

In this study, the combination of CA and RS frameworks will focus on optimization of mobile Web caching performance. CA is useful to store and update the knowledge while RS is used to find the most significant attributes and simultaneously to generate the decision rules. The similarity of CA and RS are in searching for the knowledge based frameworks. The proposed framework consists of the group problem solving of social intelligence, RS technique in dealing uncertainty data, and reliability of rules for multiple knowledge. While in the previous framework, the RS and CA approach are being treated individually.

3. The Proposed Rough Cultural Algorithm (RCA)

In this section, we present RCA for optimization of prefetcher and proxy cache. Figure 6 shows the proposed framework of RCA with two main components from CA; population and belief space. In the Belief Space, we implement the RS framework with three main modules; consistency analyzer, rough classifier, and induction engine. These modules are used to generalize the multiple knowledge of client and integrate it into the current belief structure.

RST is executed to classify the client behavior and mobile event specification, and consequently analyzing the historical data of cache or not cache for dataset reduction. Accordingly, CA is employed to solve optimization problems for the Population Space to enclose an evolving population of individuals. Meanwhile, the Belief Space will be exploited to record regions of each population parameter for high performance of population members.

The representation [23] of these individuals is considered as chromosomes that are used in evolutionary computation. An evolutionary computation algorithm is chosen due to its effectiveness in evolving solutions using the form of the Population Space chromosomes. One or more reproductive operators create the individuals of the next generation from the chromosomes of the current generation's individuals. Each operator is a function that takes a specified number of chromosomes and outputs a specified number of chromosomes. After the newly created individuals are evaluated with performance function, an acceptance function determines which of them can influence the group's problem solving experience in the Belief Space.

The Belief Space [23] of the CA contains a representation of global knowledge about the members of the population. This global knowledge consists of generalizations of the experience of selected members of the Population Space. The generalizations are beliefs about the properties of individuals that are successful or not in solving a problem. Subsequently, the RS receives the individual information and discards the data with redundant information. The rough classifier calculates the equivalence class for the knowledge representation, and finally, the induction engine decreases the rules. As a result, the reliable rules are produced as output for the Belief Space.

These rules or procedures generalize an individual's experience and will be integrated into the current belief structure. This belief structure incorporates the experience of the accepted individuals and will be adjusted according to the basis of the new information; the revised beliefs will manipulate the search for high performing individuals in future population generations.

In addition, the Communications Channel Protocol [23] determines the structures of the Belief Space and Population Space that influence the other structures, and in which order.

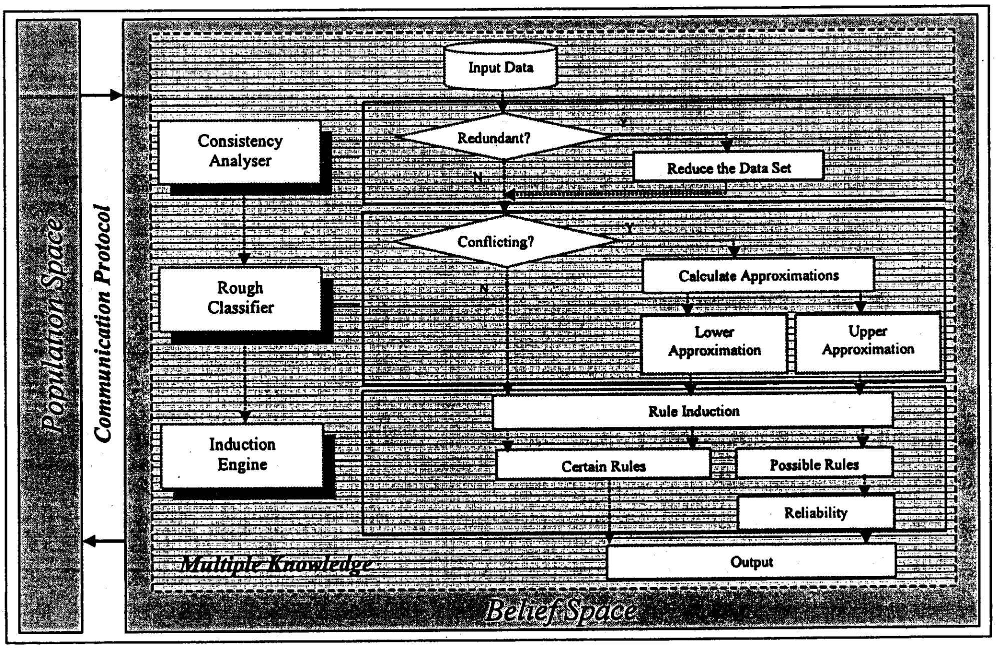

Figure 6: Proposed Framework of Rough Cultural Algorithm.

4. Conclusion

In this paper, we proposed a framework of RCA for performance enhancement of mobile Web caching. The proposed framework is based on RS as a knowledge representation for uncertainty in data, and CA for optimizing the performance of proxy caching that use to store the knowledge between each generation. In this research, we will consider the performance enhancement of mobile web caching based on the client behavior and the event specification including resource, mobility and environment events.

The issues to be solved in this study are the learning task that will require explicit representation which deals with uncertainty. The employed evolutionary learning methods must be able to work with such representation. In this phase, we will look for ways to represent uncertainty in developing rules. Subsequently we will investigate how this uncertain knowledge can be exercised directly to evolutionary prefetching and learning. Hence, our goal in this study is to find a suitable knowledge representation for mobile data, and consequently develop a feasible framework that combines the representation with the appropriate evolution method.

In the future, we plan to study how to apply the CA to deal with multiple knowledge of mobile Web caching in the Belief Space for selected members of the Population Space. We hope that our proposed method will capitulate significant contributions in reducing latency of mobile network.

Acknowledgment

This work is supported by MOSTI and RMC, Universiti Teknologi Malaysia, Skudai, Johor Bahru, MALAYSIA. Authors would like to thank Bio-Cache and Soft Computing Research Group (SCRG), Faculty of Computer Science and Information System for their continuous support and devotion in making this study a success.

References

[1] Curran, K., & Duffy, C. (2005). Understanding and Reducing Web Delays. *Int. J. Network Mgmt*, 15, 89-102.

[2] Caching Tutorial for Web Authors http://www.web-caching.com/mnot_tutorial/intro.html

[3] Saiedian, M., & Naeem, M. (2001) Understanding and Reducing Web Delays. *IEEE Computer Journal*, December 2001:34(12).

[4] Dynamic Caching, foedero.com, http://www.foedero.com/dynamicCaching.html

[5] Fan, L., Jacobson, Q., Cao, P., & Lin, W. (1999). Web prefetching between low-bandwidth clients and proxies: potential and performance. *Proceedings of the 1999 ACM SIGMETRICS International Conference on Measurement and Modelling of Computer Systems, Atlanta, Georgia, USA,* pp 178–187.

[6] Santhanakrishnan, G, Amer, A., & Chrysanthis, P.K.(2005). Towards Universal Mobile Caching, *Proceedings of MobiDE'05, Baltimore, Maryland, USA*, pp.73-80.

[7] Ari, I., Amer, A., Gramacy, R., Miller, E. L., Brandt, S., & Long, D. D. E. (2002). Adaptive Caching using Multiple Experts. In *Proc. of the Workshop on Distributed Data and Structures*.

[8] Wu, S., Chang, C., Ho, S., & Chao, H.(2007). Rule-based intelligent adaptation in mobile information systems, *Expert Systems with Applications*, DOI:10.1016/j.eswa.12.014.

[9] Komninos, A. & Dunlop, M.D. (2007). A calendar based Internet content pre-caching agent for small computing devices, *Pers Ubiquit Comput*, DOI 10.1007/s00779-007-0153-4

[10] Conrad, M. (1975). Analyzing ecosystem adaptability. *Mathematical Biosciences*, vol. 27, pp. 213-230.

[11] Conrad, M. (1978). Evolution of adaptive landscape. *Theoretical Approaches to Complex Systems* (R. Heim and G. Palm, eds.), vol. 21 of *Springer Lecture Notes in Biomathematics*, pp. 147-169, Springer-Verlag.

[12] Russell, S. J. & Norvig, P. (1995). *Artificial Intelligence a Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey.

[13] Zadeh, L. (1965). Fuzzy sets. *Information and Control*, Vol. 8, pp. 338-353.

[14] Pawlak, Z. (1991). *Rough Sets - Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers.

[15] Reynolds, R. G., & Chung, C. (1996). *A Test Bed for Solving Optimization Problems Using Cultural Algorithms, in Evolutionary Programming V*, John R. McDonnel, and Peter Angeline, Editors, A Bradford Book, MIT Press, Cambridge Massachusetts, pp. 225-236.

[16] Fogel, D.B. (1995). *Evolutionary Computation - Toward a New Philosophy of Machine Learning*. IEEE Press.

[17] Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc.

[18] Reynolds, R.G. (1994). An introduction to cultural algorithms *Proceedings of the Third Annual Conference on Evolutionary Programming, River Edge, NJ* (A. V. Sebald and L. J. Fogel, eds.), pp. 131-139, World Scientific Publishing.

[19] Wakaki,T., Itakura, H.,Tamura, M., Motoda, H., & Washio, T. (2006). A Study on Rough Set-Aided Feature Selection for Automatic Web-page Classification, *Web Intelligence and Agent Systems: An International Journal 4*, pp.431–441, IOS Press.

[20] Lazar, A. & Reynolds, R. G. (2002). Heuristic Knowledge Discover for Archaeological Data Using Cultural Algorithms and Rough Sets, in *Vol. 2: Heuristics and Optimization for Knowledge Discovery* (ed. Ruhul A. Sarker, Hussein A. Abbass, and Charles S. Newton) by Idea Group Publishing, USA.

[21] Liang, A.H., Maguire, B. & Johnson, J. (2000). *Rough Set WebCT Learning*. Springer-Verlag Berlin Heidelberg, pp.425-436.

[22] Ngo, C. L. & Nguyen, H. S. (2004). *A Tolerence Rough Set Approach to Clustering Web Search Results*. Springer-Verlag Berlin Heidelberg, pp.515-517.

[23] Cowan, G.S. & Reynolds, R.G. (2003). *Acquisition of Software Engineering Knowledge.* World Scientific Press, Vol.14, Singapore, pp.55-61.

[24] Triantaphyllou, E. & Felici, G. (2006). *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*, Massive Computing Series, Springer, Heidelberg, Germany, pp.359-394.

[25] Peng, B & Reynolds, R. G. (2004). Cultural Algorithms: Modeling of How Cultures Learn to Solve Problems. *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004)*, IEEE Computer Society.